VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra telekomunikační techniky

Autentizace uživatelů pomocí protokolu Diametr a Radius

User Authentation with Diameter and Radius Protocol

2018 Bc. Lukáš Janča

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra telekomunikační techniky

Zadání diplomové práce

Student: **Bc. Lukáš Janča**

Studijní program: N2647 Informační a komunikační technologie

Studijní obor: 2601T013 Telekomunikační technika

Téma: Autentizace uživatelů pomocí protokolu Diametr a Radius
User Authentication with Diameter and Radius Protocol

Jazyk vypracování: čeština

Zásady pro vypracování:

Autentizace uživatelů je velmi důležitá součást bezpečného přístupu do sítě. Cílem diplomové práce je navrhnout autentizaci uživatelů s využitím protokolů Radius a Diametr v prostředí virtuálních serverů.

Řešení práce musí splňovat následující body:
1. Studium a popis protokolu Radius.
2. Studium a popis protokolu Diametr.
3. Návrh řešení autentizace a autorizace uživatelů.
4. Analýza a porovnání obou autentizačních protokolů.

Seznam doporučené odborné literatury:

[1]HASSELL, Jonathan. *Radius*. O'Reilly Media 2002. ISBN-13: 978-0596003227
[2]NAKHJIRI,Mahsa, NAKHJIRI, Madjid. *AAA and Network Security for Mobile Access: Radius, Diameter, EAP, PKI and IP Mobility*. Wiley; 1 edition 2005. ISBN-13: 978-0470011942

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **Ing. Pavel Nevlud**

Datum zadání: 01.09.2016
Datum odevzdání: 30.04.2018

VYSOKÁ ŠKOLA BÁŇSKÁ-TECHNICKÁ UNIVERZITA Fakulta elektrotechniky a informatiky 1D16 OSTRAVA

doc. Ing. Miroslav Vozňák, Ph.D.
vedoucí katedry

prof. Ing. Pavel Brandštetter, CSc.
děkan fakulty

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 30. dubna 2018 .....

Rád bych poděkoval vedoucímu práce Ing. Pavlu Nevludovi za odbornou pomoc, konzultaci a školní síťová zařízení, která mi poskytl při vytváření této diplomové práce.

Abstrakt

Tato diplomová práce se zabývá praktickým návrhem protokolů RADIUS a Diameter v prostředí virtuálních serverů. Protokoly RADIUS a Diameter zajišťují podporu procesů autentizace, autorizace a účtování v síťové infrastruktuře. Teoretická část práce obsahuje porovnání mezi protokoly z hlediska struktury zpráv, zabezpečení, spolehlivosti, rozšiřitelnosti, podpory služeb a vzájemné interakce. Praktická část zahrnuje implementaci protokolů užitím programů FreeRADIUS, freeDiameter, Hostapd, wpa_supplicant a dalších programů. Praktická implementace je plně funkční pro využití IPv4 a IPv6. Výstupem práce jsou skriptovací soubory, kterými lze plně sestavit obsah praktické části. Výstup dále zahrnuje zachycenou komunikaci síťového provozu. V závěru práce je provedena analýza mezi protokoly z hlediska praktické funkčnosti, úrovně zabezpečení, zpětné kompatibility a možnosti rozšiřitelnosti.

Klíčová slova: RADIUS; Diameter; AAA protokoly; virtuální servery; autentizace; autorizace

Abstract

This diploma thesis deals with the practical design of RADIUS and Diameter protocols in the environment of virtual servers. RADIUS and Diameter protocols provide support for authentication, authorization, and accounting procedures within the network infrastructure. The theoretical part of the thesis contains a comparison between protocols in terms of message structure, security, reliability, extensibility, service support and mutual interaction. The practical part includes the implementation of protocols using FreeRADIUS, freeDiameter, Hostapd, wpa_supplicant and other software. The practical implementation is fully functional for IPv4 and IPv6 usage. The output of the thesis includes scripting files, which can fully compile the content of the practical part. Furthermore, the output includes captured network traffic. Conclusion of thesis contains performed analysis between protocols in terms of practical functionality, level of security, backward compatibility and possibility of extension.

Key Words: RADIUS; Diameter; AAA protocols; virtual servers; authentication; authorization

Obsah

Seznam použitých zkratek a symbolů

AAA	–	Authentication, Authorization and Accounting protocol – Autentizační, autorizační a účtovací protokol
RADIUS	–	Remote Authentication Dial In User Service – Uživatelská vytáčená služba pro vzdálenou auntentizaci
SIM	–	Subscriber Identity Module – Účastnická identifikační karta
MITM	–	Man-in-the-middle – Člověk uprostřed
WLAN	–	Wireless Local Area Network – Bezdrátová lokální síť
AP	–	Acesss Point – Přístupový bod
POP	–	Point of Presence – Bod přítomnosti
NAS	–	Network Access Server – Síťový přístupový server
PPP	–	Point-to-point Protocol – Protokol bod-k-bodu
UDP	–	User Datagram Protocol – Protokol uživatelských datagramů
TCP	–	Transmission Control Protocol – Protokol kontroly přenosu
STCP	–	Stream Transmission Control Protocol – Protokol kontroly proudového přenosu
IETF	–	Internet Enginnering Task Force – Komise pro technickou stránku internetu
ASM	–	Application-Specific Module – Aplikačně-specifický modul
ASI	–	Application-Specific Information – Aplikačně-specifická informace
SSL	–	Secure Socket Layer – Zabezpečná vrstva koncových bodů
EAP	–	Extensible Authentitacion Protocol – Rozšiřitelný autentizační protokol
TLS	–	Transport Layer Security – Zabepečení přenosové vrstvy
TTLS	–	Tunneled Transport Layer Security – Tunelované zabezpečení přenosové vrstvy
PKS	–	Pre-Shared Key – Předem sdílený klíč
LEAP	–	Lightweight Extensible Authentication Protocol – Odlehčený rozšiřitelný autentizační protokol
PEAP	–	Protected Extensible Authentication Protocol – Zabezpečený rozšiřitelný autentizační protokol
RFC	–	Request For Comments – Žádost o komentáře
PAP	–	Password Autentication Protocol – Protokol autentizace heslem
CHAP	–	Challenge Handshake Authentication Protocol – Protokol autentizace pomocí výzvy
MS-CHAP	–	Microsoft version of Challenge Handshake Authentication Protocol – Protokol autentizace pomocí výzvy, verze Microsoft

TLV	–	Type-Length-Value Form – Forma typu, délky a hodnoty
ID	–	Identifier – Identifikátor
VSA	–	Vendor-Specific Attribute – Výrobcem-specifický atribut
MD5	–	Message-Digest 5 – Strávení-zprávy, verze 5
IP	–	Internet Protocol – Internetový protokol
DHCP	–	Dynamic Host Configuration Protocol – Protokol konfigurace hostitelského počítače
QoS	–	Quality of Service – Technologie kvality služeb
IPv4	–	Internet Protocol version 4 – Internetový protokol, verze 4
IPv6	–	Internet Protocol version 6 – Internetový protokol, verze 6
SNMP	–	Simple Network Management Protocol – Jednoduchý protokol správy sítě
COPS	–	Common Open Policy Service Protocol – Protokol společné otevřené politiky služeb
AVP	–	Attribute-Value Pair – Pár atribut-hodnota
IPSec	–	Internet Protocol Security – Zabezpečení internetového protokolu
DNS	–	Domain Name System - Systém doménových jmen
FQDN	–	Fully Qualified Domain Name – Plně specifikované doménové jméno
MAC	–	Media Access Control – Přístup k médiu
CD	–	Compact disc – Kompaktní disk
SQL	–	Structured Query Language – Strukturovaný dotazovací jazyk
LDAP	–	Lightweight Directory Access Protocol – Odlehčený protokol pro přístup k adresářům
SSH	–	Secure Shell – Zabezpečená schránka
FTP	–	File Transfer Protocol – Protokol přenosu souborů
PAM	–	Pluggable Authentication Module – Zásuvný autentizační modul
Hostapd	–	Host Access Aoint Daemon – Démon přístupového bodu hostitele
GPLv2	–	General Public License version 2 – Všeobecná veřejná licence, verze 2
SHA	–	Secure Hash Algorithm – Bezpečný hašovací algoritmus
DNS	–	Domain Name System – Systém doménových jmen
VŠB	–	Technical University of Ostrava – Vysoká škola Báňská
VUT	–	Brno University of Technology – Vysoké učení technické v Brně
Wi-Fi	–	Wireless Fidelity – Bezdrátová síť
VUT	–	Wi-Fi Protected Access – Chráněný přístup k Wi-Fi
FTPS	–	FTP Secure – FTP s podporou SSL/TLS

Seznam obrázků

Seznam tabulek

Seznam výpisů zdrojového kódu

Úvod

Ve světě počítačové techniky existuje mnoho hrozeb nejen ve smyslu počítačových virů ale i jiných narušení bezpečnosti. Mezi četná narušení bezpečnosti patří nedbalost a neinformovanost uživatelů či záměrné útoky na uživatelskou identitu a údaje s cílem zneužití a krádeže. Uvažujeme-li tedy o návrhu vlastní síťové infrastruktury, je důležité neopomenout určitá bezpečnostní opatření.

Pouze několik let zpátky bylo běžnou praxí nejprve navrhnout základní síťovou funkčnost a v poslední řadě zajistit síťové zabezpečení za pomoci experta na kryptografii. Tento přístup záplatování vedl k prodlevám v nasazení a navyšování finančních nároků. Lidé si začali uvědomovat, že investice do řádných zabezpečovacích opatření je vážnou záležitostí. Tímto trendem se dostalo pozornosti rozrůstajícímu se oboru síťového zabezpečení. Bylo vyvinuto několik kryptografických konceptů a algoritmů, které daly vzniku novým síťovým protokolům. I přes tento vývojový úspěch bylo stále obtížné spravovat síť, která generuje služby a příjmy.

Přístupové sítě musí být nějak kontrolovány a uživatelé společně se zařízeními, kteří se pohybující v rozsáhlé síti, musí prokázat svou totožnost. Taktéž musí být zajištěno přiřazení oprávnění k využití síťových služeb. Dalším požadavkem na přístupové sítě bylo zachytit síťovou aktivitu uživatele, například pro účely vyúčtování. Tyto nároky daly vzniku nové rodině síťových protokolů pro autentizaci, autorizaci a účtování (AAA). Evolucí protokolové rodiny AAA je poskytnout funkce k vykonávání úloh od správy prostředků až k zajišťování mobility. Současným požadavkem na síťové infrastruktury je umožnit uživatelům připojení přes výběr různých rozhraní, zařízení a technologií.

Mezi dva prominentní protokoly z rodiny AAA patří Uživatelská vytáčená služba pro vzdálenou autentizaci (RADIUS) a jeho rozšířený následník protokol Diameter. Tyto zmíněné protokoly jsou tématem mé práce a dále se na ně zaměřím.

1 Autentizace, autorizace a účtování

Model AAA vychází z tří základních procesů, které se provádí při žádosti o přístup do sítě. Prvním procesem je ověření identity připojujícího se uživatele či zařízení a takto nazýváme autentizaci. Druhým krokem je přiřazení oprávnění k užití služeb nebo procesů sítě již autentizovaných uživatelů. Toto přiřazení oprávnění se označuje jako autorizace. Účtování je proces zaznamenání aktivity uživatele, který byl připuštěný do sítě.

Účelem užití AAA procesů při konstrukci sítě je vypomoci síťovému poskytovateli a jeho zákazníkům proti podvodu, útokům, nevhodnému využití prostředků či ztrátě na příjmech. Tato kapitola se týká popisu jednotlivých AAA procesů, odůvodnění jejich vzájemného propojení a modelu obecné AAA architektury. [1, 2]

1.1 Autentizační koncepty

Pod pojmem autentický označujeme věc, která není falešná či imitovaná, ale můžeme ji považovat za pravou. Proces autentizace se skládá ze dvou aktů a to poskytnutí důkazu pravosti informace a její verifikace na základě předložených důkazů. Existují tři základní typy autentizace a jejich výpis je součástí této kapitoly. [2]

1.1.1 Uživatelská autentizace

Pod tímto typem rozumíme uživatele, který si přeje získat přístup do sítě a předkládá své doklady. Tyto doklady jsou poté verifikovány samotnou sítí pro ověření identity uživatele. [2]

Pojmem uživatel se v AAA kontextu může označovat osoba i zařízení. Označení klient má v AAA jiný význam a bude vysvětleno v kapitole 1.1.4.2. Obecně existuje rozdíl v autentizačním postupu, zdali síť obsluhuje zařízení nebo člověka. První případ může nastat, když osoba zadává své údaje do pevného terminálu přistaveného vlastníkem sítě. Kritériem autentizace jsou pouze osobní údaje a použitý terminál není ověřován. Druhým případem je buňková rádiová síť. Požadavek k využití služeb sítě je nyní ověřován vůči údajům uloženým na účastnické identifikační kartě (SIM) a autentizováno je pouze zařízení.

Tyto rozdíly hrají roli při využití sítě, definici bezpečností politiky a způsobu uložení autentizačních dokladů. [2]

1.1.2 Autentizace zpráv

Uživatelská autentizace rozhoduje pouze o legitimnosti koncových bodů síťové komunikace. Účelem autentizace zpráv je zajištění a verifikace integrity jednotlivých dat. Provedením autentizace zpráv je příjemce ujištěn, že zpráva přichází z pravého zdroje a nebylo s přenesenou zprávou manipulováno. Z tohoto důvodu se autentizace zpráv považuje za mechanismus ochrany integrity.

Ochrana integrity má za účel zabránit škodlivému a zamýšlenému poškození dat takzvaným člověkem uprostřed (MITM), který manipuluje s obsahem zprávy. Tímto se ochrana liší od

cyklického redundantního součtu a informačních teoretických kódů, které jsou navrženy pro zmírnění přirozeného a náhodného poškození dat při přenosu na fyzickém médiu. Mezi metody zajištění autentizace zpráv patří symetrická kryptografie a digitální certifikáty. [2]

1.1.3 Vzájemná autentizace

Uživatelskou autentizací prokazuje svou identitu pouze jeden koncový bod síťové komunikace. Dává smysl provést autentizaci na obou koncových bodech před sestavením komunikace. S navýšením počtu bezdrátových lokálních sítí (WLAN) je nyní vhodné autentizovat i přístupový bod (AP) vůči uživateli přistupující do sítě. [2]

1.1.4 Modely pro výměnu zpráv

1.1.4.1 Model autentizace dvou stran

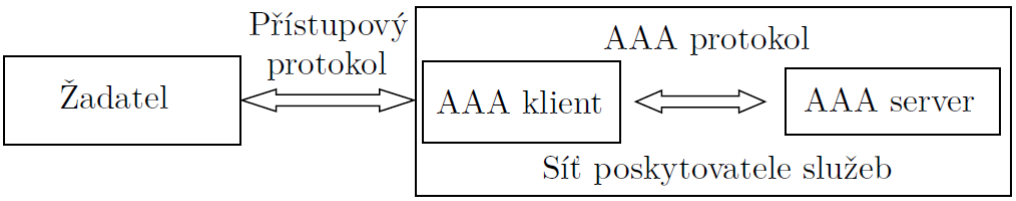
Tento způsob značí interakci dvou rovnocenných účastníků (angl. peers) na přímé komunikační lince bez využití propojujících uzlů jako brán (angl. gateway) či zástupců (angl. proxy). Význačný případ je přímá autentizace uživatele a serveru.

1.1.4.2 Model autentizace tří stran

S navyšováním velikosti sítí a počtu připojujících se uživatelů došlo k trendu nasazování specifického bodu přítomnosti (POP). Jedná se o jednoduché nízkonákladové zařízení s omezenou výpočetní schopností a malou databází. POP typicky interaguje přímo s uživatelem, ale dotazuje se centrálního serveru na úkoly a rozhodnutí týkající se uživatele. POP nejsou obecně uzpůsobeny k výkonu autentizačních procesů, a proto byl model autentizace rozšířen pro tři strany:

1. *Žadatel*: Je uživatel žádající přístup do sítě za účelem využití služeb poskytovatele.
2. *AAA klient*: Je hraniční zařízení sítě interagující s žadatelem. V AAA kontextu se tato entita nazývá síťový přístupový server (NAS).
3. *AAA server*: Je zařízení se skutečnou autoritou a potřebnou databází informací k učinění rozhodnutí týkající se udělení přístupu žadatele.

Model autentizace tří stran je vyznačen na obrázku 1.1.1.



Obrázek 1.1.1: Model autentizace tří stran v AAA architektuře [2]

1.1.5 AAA protokoly pro autentizaci zpráv

V malých sítích je možné konfigurovat AAA klienta a AAA server do společného úložiště. Tento přístup není praktický pro větší či vícedoménové sítě s velkým počtem POP, které zastupují funkci NAS a autentizace se provádí dle modelu tří stran. Přirozeně platí, že autentizační výměna mezi koncovými body odpovídá komunikaci mezi žadatelem a AAA serverem, ale NAS se taktéž účastní ve výměně autentizačních zpráv.

NAS se typicky chová jako bod dělení protokolů, jelikož komunikace směrem od NAS k AAA serveru je většinou vedena přes privátní, drátovou a důvěryhodnou část sítě. Komunikace od NAS k žadateli je provozována na nedůvěryhodném a často bezdrátovém médiu. Za účelem zajištění interoperability mezi různými síťovými zařízeními bylo pro jednotlivé segmenty modelu autentizace tří stran standardizováno několik protokolů. [2]

1.1.5.1 Žadatel-AAA server komunikace

V kapitole 1.1.5.1 bylo řečeno, že neexistuje přímá komunikace mezi žadatelem a AAA serverem. V dnešní době nejrozšířenější AAA protokol, RADIUS, byl navržen tak, aby NAS umožnil předat žadatelův požadavek a jeho doklady k AAA serveru a následně přenést odpověď AAA serveru zpátky k žadateli. Struktura zpráv Žádost-Přístupu (angl. Access-Request) a Výzva-Přístupu (angl. Access-Challenge) v RADIUS protokolu značí, že byla navržena pro autentizaci založené na heslech.

1.1.5.2 NAS-AAA server komunikace

Pro komunikaci mezi NAS a AAA serverem je uzpůsoben RADIUS i DIAMETER. Původním předpokladem byla existence pouze jednoho skoku, ale možné je využít AAA zástupců pro komunikaci mezi NAS a AAA serverem. Je vhodné zmínit, že užitím zástupců může dojít k přenosu informace v neprivátních sítích a je tedy nutné zajistit zvláštní bezpečností opatření.

1.1.5.3 Žadatel-NAS komunikace

Komunikace mezi žadatelem a NAS je typicky vedena jedním skokem takzvanou přístupovou linkou. Přístupová linka zajišťuje fyzický kanál a protokol linkové vrstvy. K zajištění formátování

paketů, rámcování a mechanismů vícenásobného přístupu slouží protokol bod-k-bodu (Point-to-point protocol). Bezdrátové přístupové technologie jako 802.11 WLAN mají své vlastní rámcové mechanismy a nevyžadují dodatečně PPP pro účel přenosu zpráv na linkové vrstvě. K umožnění služeb na síťové vrstvě je nejprve nutné zajistit autentizaci, a proto výměna zpráv mezi žadatelem a NAS musí být provedena užitím protokolu přístupové technologie linkové vrstvy. [2]

1.2 Autorizace

Autorizace je definovaná jako proces určení, zdali může být určité oprávnění uděleno žadateli s konkrétními doklady. Může se jednat o oprávnění přístupu k zdroji jako komunikační linky, informační databáze, či jiného vlastnictví poskytovatele služeb.

V komerčním užití obecně autorizace slouží k ochraně příjmů nebo přiražení oprávnění k některé službě. Jako příklad nutnosti autorizace lze uvést případ poskytovatele audiovizuálních služeb. K ověření žadatelovy žádosti o přístup k audiovizuálnímu materiálu ve vyšší kvalitě musí existovat způsob kontroly uživatelského profilu, zdali byl uhrazený poplatek za prémiové služby. Dále je nutné kontaktovat entity kontrolující hodnotu přenosové rychlosti pro konečné autorizování žadatele. [2]

1.2.1 Autorizační politika

K zajištění konzistence a škálovatelnosti procesu autorizace je často nutné nastavit sadu zásad neboli politiku. Protože existuje mnoho typů politik, jako například bezpečností politika, skupinová příslušnost či roamingová politika, je důležité ustanovit politický rámec. Politický rámec definuje různé prvky architektury, jako je repositář zásad a politiku rozhodovacích bodů. Repositář zásad typicky zahrnuje informace o dostupných službách, zástupcích k rozhodnutí o autorizaci, zásadách k autorizačním rozhodnutím a záznamech o autorizačních událostech.

Politický rámec taktéž definuje procesy k správě a sdílení informací s jinými entitami v síti. Občas je nutné, aby AAA server interagoval se síťovými entitami za účelem učinění odpovídajícího autorizačního rozhodnutí. Tato interakce musí AAA serveru umožnit načíst politiku a vynutit zásady při autorizačním procesu. [2]

1.3 Účtování

Účtování obecně slouží k záznamu síťových prostředků a služeb, které v síti využívá uživatel. Uložený záznam o aktivitě je převážně využitý k fakturaci, ale existuje i řada jiných využití. Mezi různé užití patří například:

1. *Audit*: Je akt verifikace správnosti dokumentů předložených poskytovatelem služeb nebo dodržování zásad použití.

2. *Přidělení nákladů*: Značí vyhodnocení struktury spojené například s datovou a hovorovou částí telefonie.

3. *Analýza trendů*: Slouží pro stanovení prognózy budoucího využití pro účely plánování kapacity.

Každé z užití může být zpracováno jinou entitou logického řízení. Sbírané informace se obecně označují za účtovací data nebo účtovací metriky. Každé ze zmíněných užití může mít různé nároky na zabezpečení a spolehlivost účtovacích dat. [1, 2]

1.3.1 Zabezpečení účtování

V rámci účtování dochází k přenosu dvou typů komunikačních dat. První typ je výměna účtovací politiky a druhý typ je sběr účtovacích záznamů. Podkladem pro fakturaci jsou účtovací záznamy a existuje tedy vysoké riziko jejich případného podvržení.

Mezi bezpečnostní opatření patří zamezení práva čtení a modifikace účtovací politiky a záznamů neautorizovanými entitami. Integritu a pravost zdroje účtovacích dat lze zajistit uplatněním digitálních certifikátů. Samotnou verifikaci generovaných účtovacích záznamů je možné provést důvěryhodnou třetí stranou či obecnou entitou zajišťující auditové služby. [2]

1.3.2 Spolehlivost účtování

Typické účtovací selhání je způsobenou paketovou ztrátou, výpadkem sítě a restartováním serveru. Je tedy důležité, aby systémy řízení účtování byly škálovatelné a spolehlivé. Účtovací služby většinou požadují, aby NAS odesílal AAA serveru zprávy značící začátek a konec účtovací relace. Aby bylo zamezeno ztrátě zprávy značící konec relace, která nese velké množství informací, je možné využít takzvaného průběžného účtování.

Procedura průběžného účtování zajišťuje periodické aktualizace nesoucí informace o účtovací relaci. Tímto způsobem je možné na základě záchytných bodů provést rekonstrukci záznamu relace i v případě selhání při přenosu.

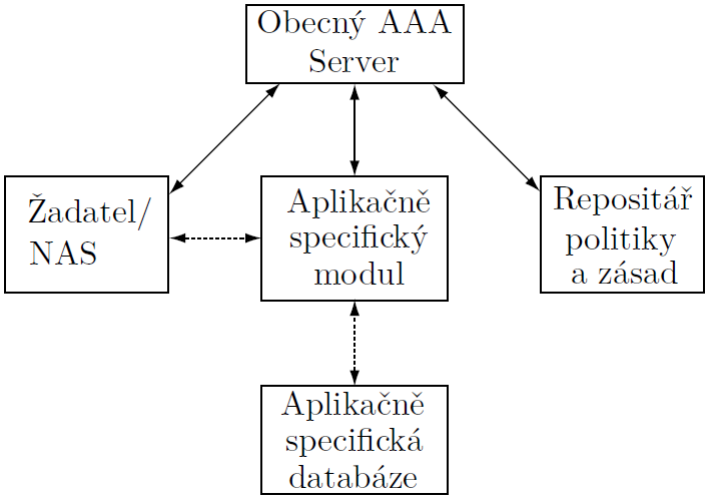
Účtování se často provádí na transportním protokolu uživatelských datagramů (UDP). Při implementaci UDP je nutné počítat s neprováděnou kontrolou správnosti doručení zaslaných paketů. Je možné uplatnit mechanismů opakovaného zaslání přímo na UDP nebo využít spolehlivějšího protokolu kontroly přenosu (TCP) či protokolu kontroly proudového přenosu (SCTP). [2]

1.4 Obecná AAA architektura

V kapitole 1.1.4.2 byl vysvětlen model autentizace tří stran a z něj vychází obecná AAA architektura doplněná několika moduly. Komise pro technickou stránku internetu (IETF) vytvořila doporučení RFC 2903 [3], které specifikuje interakci obecné AAA architektury a jiných entit síťového řízení.

Každá z těchto entit síťového řízení poskytuje specifikou službu či funkci. Entity síťového řízení jsou označovány jako AAA aplikace. Příklady AAA aplikací jsou řízení přenosové rychlosti,

kvalita služeb a služby mobility. Protokol RADIUS, na rozdíl od novějšího protokolu Diameter, nebyl navržen pro tuto obecnou AAA architekturu. Protokol Diameter má oddělené specifikace pro každou z AAA aplikací. RFC 2903 definuje nový koncept zvaný aplikačně-specifický modul (ASM), které abstraktně popisuje funkčnost entity, s kterou je AAA server v interakci.



Obrázek 1.4.1: Obecný model pro interakci AAA serveru a různých řídicích entit [2]

Obrázek 1.4.1 zobrazuje interakci mezi AAA serverem a různými řídicími entitami dle modelu obecné AAA architektury na základě doporučení RFC 2903. Kromě ASM musí AAA server spolupracovat i repositářem politiky a zásad k získání pravidel, které se týkají konkrétní aplikace a uživatele. Protokoly využívané k interakci mezi uživatelem, AAA serverem a repositářem se mohou lišit od protokolů pro interakci mezi AAA serverem a ASM. Tato odlišnost může taktéž platit mezi ASM a jeho databází v případě, kdy nejsou oba moduly umístěny společně.

Je důležité zmínit, že každá ze služeb nese aplikačně-specifickou informaci (ASI), jako například hodnotu přenosové rychlosti, které porozumí pouze ASM. Z tohoto důvodu se AAA server při autorizaci odkáže přímo na ASM. V textu později uvidíme, že většina AAA protokolů pracuje se specifickými daty zvanými atributy, které jsou navrženy pro každý z typů ASI. AAA server dokáže na základě typu atributu rozlišit, ke které aplikace se vztahuje. Následující případ popisuje řetězec akcí, které by se měly provést v obecné AAA architektuře:

- Uživatel nebo síťové zařízení obsluhující uživatele (NAS) zašle kombinovaný autentizační požadavek a žádost o službu na AAA server. Pro schválení žádosti se od uživatele očekává poskytnutí platných dokladů.
- AAA server verifikuje doklady odkázáním se uživatelskou databázi či repositář zásad. AAA server po úspěšné autentizaci dále kontroluje obsah autorizačního požadavku a určí jeho účel.

- AAA server provádí autorizační rozhodnutí pro jednotlivé atributy požadavku buď předáním požadavku na ASM, dotazem na repositář zásad nebo přesměrováním na jiný AAA server mimo administrativní doménu.
- AAA server informuje síťové zařízení (NAS) o autorizačním rozhodnutí a popřípadě poskytne nutné informace k přípravě služby vyžádané uživatelem. AAA server může dále nařídit POP, aby došlo k záznamu účtovacích dat.

Privátnost a zabezpečení některé z informačních komponent musí zajistit AAA server procesem šifrování či autentizace při přenosu. V případech autorizace je vždy konečné rozhodnutí prováděno AAA serverem, i když jsou služby, síťové zdroje, politika rozhodnutí a síťová správa prováděny jinými entitami. [2]

2 Autentizační mechanismy

Základní definice a různé typy autentizace jako uživatelská autentizace a autentizace zpráv byly popsány v kapitole 1. S nasazováním neustále se rozvíjejících sítí a pokročilých služeb se dnes potýkáme s útočníky, kteří k napadení využívají stále silnější výpočetní zařízení. Není tedy žádné překvapení, že došlo k vývoji a standardizaci mnoha autentizačních mechanismů, modelů a procedur. Smyslem této kapitoly je poskytnout pochopení některých z autentizačních mechanismů, které jsou běžně využívány v sítích. [2]

2.1 Třídy autentizačních mechanismů

Problém autentizace je velmi starý a list možných autentizačních mechanismů by mohl být příliš dlouhý. Pro tento důvod je vhodnější poskytnout klasifikaci celkem sedmi tříd, které zahrnují téměř všechny využívané autentizační mechanismy. Mezi třídy autentizačních mechanismů patří:

1. *Pevné heslo*: Jedná se o jednoduchou metodu uživatelské autentizace, kdy uživatel poskytne pár uživatelského jména a hesla společně s autentizačním požadavkem či žádostí o přístup do sítě. Požadavek je zpracován serverem dohledáním uživatelských údajů v databázi. Heslo je většinou zasláno šifrované předem určeným typem hašovací funkce (jednosměrný převod vstupních dat). Mezi problémy této metody patří odposlech, hádání hesla a slovníkové útoky. K zabránění slovníkových útoků je možné využít takzvané metody kryptografické soli. Hašovací funkce se provádí na uživatelském heslu v kombinaci s náhodnými bity (sůl) a zaručí tak odlišný výstup hašovací funkce pro dva uživatele s totožným heslem.

2. *Jednorázové heslo*: Mechanismus využití hesla pro jediné použití byl vyvinut s cílem vyřešit problém opakovaných útoků. Klient je buď vybaven sadou jednorázových hesel, nebo kartou schopnou vypočítat heslo na základě předurčeného algoritmu. Existuje taktéž možnost sloučení pevného hesla a náhodných znaků, které generuje karta pro určitou časovou dobu.

3. *Výzva/Odpověď*: Autentizační mechanismus výzva/odpověď je založen na předpokladu, že uživatel použije vlastní heslo či více pokročilou metodu k výpočtu odpovědi na výzvu obdrženou serverem. Výzva je typicky hodnota generována s velkou náhodností. Odpověď se vypočítá hašovací funkcí sloučené výzvy s heslem. Důvodem vytvoření mechanismu bylo zabránit odposlechům přenášeného hesla v pevné podobě a omezit opakované útoky při velké míře neurčitosti náhodných hodnot výzvy.

4. *Anonymní výměna klíčů*: Podstatou anonymní výměny klíčů je zaslání veřejných klíčů mezi dvěma účastníky za účelem získání sdíleného hesla. Samotné klíče a kryptografické metody jsou pouze povinné k prvotní výměně klíčů a nikoliv skutečné autentizaci. V základní formě mechanismu účastníci nepředkládají důkaz své identity a výměna je považována za

anonymní. Anonymní výměna je náchylná na útok typu MITM, ale k ochraně lze využít důvěryhodnou autoritu k podpisu klíčů. Tato metoda se nazývá digitální certifikát, který zahrnuje oba veřejné klíče a podpis důvěryhodné autority.

5. *Důkaz s nulovou znalostí hesla*: Tento mechanismus je navržen k řešení autentizačního problému vyžadující předem sdílených klíčů či hesel, které čelí riziku odposlechu nebo napadení MITM. Princip je prokázání znalosti klíče jednou stranou vůči straně druhé bez nutnosti informaci zaslat nezabezpečeným kanálem. Tento mechanismus je silně patentován a nezískal rozšířenou popularitu.

6. *Serverové certifikáty a uživatelská autentizace*: Předpokladem tohoto mechanismus je navázání bezpečného kanálu mezi dvěma účastníky založeného na jednosměrné autentizaci (pouze jednoho účastníka k druhému). Autentizace druhého účastníka může proběhnout přes zavedený bezpečný kanál a to méně důmyslným způsobem. Příkladem je nejprve ověření serverové části prokázáním certifikátu vůči uživateli a autentizace uživatele probíhá dodatečně po již ustanoveném bezpečném kanálu. Myšlenka na poskytování serverových certifikátů získala popularitu s příchodem zabezpečené vrstvy koncových bodů (SSL) k účelu nakupování na internetu.

7. *Vzájemná autentizace digitálním certifikátem*: Nejvíce robustní a bezpečný mechanismus pro vzájemnou autentizaci je vyžádat server i uživatele k předání svých digitálních certifikátů mezi sebou. Účastníci nemusí spoléhat na předchozí domluvu či předem sdílené klíče k ustanovení spojení. Správa certifikátů je však složitá záležitost vyžadující pokročilé znalosti z hlediska úschovy klíčů.

Smyslem klasifikace bylo nastínit klady a zápory při návrhu a praktické implementaci různých autentizačních mechanismů. Klasifikace je seřazena od nejméně k nejvíce bezpečným autentizačním mechanismům. [2]

2.2 Obecný autentizační mechanismus

Myšlenkou obecného autentizačního mechanismu je poskytnout rámec, který účastníkům umožní podporu pro mnoho různých autentizačních mechanismů, které lze při autentizaci společně vyjednat. Tato myšlenka je přívětivá pro síťové návrháře, jelikož snadněji umožňuje vylepšit řízení přístupu a bezpečností postupy bez potřeby zavádění změn v síťové architektuře. Svobodné vyjednání autentizačního mechanismu mezi účastníky je náchylné na takzvaný sestupný útok. Sestupný útok značí případ, kdy útočník zachytí vyjednávání a přesvědčí účastníky k zvolení slabšího a méně bezpečného autentizačního mechanismu. [2]

2.3 Rozšiřitelný autentizační protokol

Označení rozšiřitelného autentizačního protokolu (EAP) je předním příkladem obecného autentizačního rámce. Obliba EAP vychází z nátlaku na návrháře sítí a správce, aby zajistili podporu

starších autentizačních mechanismů užívaných stávající platformou a zároveň umožnili využít nových bezpečnějších autentizačních mechanismů.

Hlavní výhodou EAP je jeho pružnost, kterou zajišťuje autentizačnímu modelu tří stran, jenž byl popsán v kapitole 1.1.4.2. V případě vývoje nové autentizační metody není nutné zmodernizovat NAS k podpoře nové metody, jelikož EAP umožňuje AAA klientovi pouhé předání autentizačních zpráv k AAA serveru. Síťoví návrháři tedy nemusí konfigurovat podporu nové autentizační metody na všechny NAS v síti, ale aktualizují jediný AAA server. EAP sám neprovádí autentizační proces, ale poskytuje prostředky pro vyjednání výměny informací mezi žadatelem a AAA serverem. EAP se skládá celkem z čtyř zpráv, které jsou EAP žádost (od AAA serveru-/klienta k žadateli), EAP odpověď (od žadatele k AAA serveru/klientovi) nesoucí autentizační data, zatímco zprávy EAP úspěch a EAP neúspěch přenáší výsledek autentizačního procesu.

Při prvotní výměně zpráv EAP žádost a EAP odpověď dochází k vyjednání autentizační metody mezi žadatelem a AAA serverem. Mezi některé z mechanismů užívané EAP například patří zabezpečení přenosové vrstvy (TLS), tunelované zabezpečení přenosové vrstvy (TTLS) a předem sdílený klíč (PKS). EAP využívající jeden ze zmíněných mechanismů se poté nazývá EAP-TLS [4], EAP-TTLS a EAP-PKS. Firma CISCO dále vyvinula odlehčený rozšiřitelný autentizační protokol (LEAP) a zabezpečený rozšiřitelný autentizační protokol (PEAP). [2]

3 Protokol RADIUS

Protokol RADIUS (uživatelská vytáčená služba pro vzdálenou autentizaci) byl vytvořen samostatnou pracovní skupinou z IETF na základě potřeby umožnit autentizaci uživatele, který se připojuje do sítě za účelem využití různých služeb. V pozdějších verzích byl RADIUS doplněn taktéž podporou pro autorizační a účtovací procedury. Před nástupem protokolu RADIUS byly užívané autentizační metody velmi specifické pro konkrétní zařízení a způsobovaly mnoho režijních nákladů s minimální pružností řízení.

Účelem této kapitoly je poskytnout přehled o IETF doporučení k protokolu RADIUS z hlediska struktury zpráv, podpory autentizačních mechanismů, bezpečnostních nedostatků a spolehlivosti. [2, 5]

3.1 Základy protokolu RADIUS

RADIUS pracuje na bázi klient-server, kde je zařízení NAS označeno jako RADIUS klient. Konečný uživatel nebo zařízení, které vyžaduje autentizaci, se nazývá žadatel. Funkce a označení jednotlivých entit protokolu RADIUS tedy odpovídá modelu autentizace tří stran, který je popsán v kapitole 1.1.4.2. V procesu autentizace je RADIUS klient zodpovědný předat údaje žadatele formou požadavku vůči straně RADIUS serveru a následně vyčkat na odpověď RADIUS serveru. Na základě stanovené politiky je NAS povinen otevřít žadateli vhodný komunikační kanál. Pro účtovací procedury je NAS taktéž zodpovědný za sběr dat o využitých prostředcích a odesílání daných dat vůči RADIUS serveru.

RADIUS specifikace je předmětem několika žádostí o komentáře (RFC) neboli doporučení. Základní specifikace protokolu RADIUS byla předmětem revize (RFC 2058 a 2138) a dnes je aktuální v doporučení RFC 2865 [6]. Základní specifikace popisuje uživatelské autentizace vůči RADIUS serveru užitím protokolu autentizace heslem (PAP), protokolu autentizace pomocí výzvy (CHAP) a obdobné verze pro Microsoft (MS-CHAP), ale nepopisuje podporu pro účtování. Účtovací procedury byly samostatně standardizovány v doporučení RFC 2866. RADIUS, jakožto rozšiřitelný protokol, byl v pozdějších doporučeních taktéž doplněn podporou autentizačních metod EAP. [2]

3.2 Zprávy protokolu RADIUS

Sada zpráv protokolu RADIUS se skládá z celkem osmi jednoduchých zpráv, z kterých jsou pouze první čtyři specifikovány v základním doporučení [6]. Do seznamu zpráv patří:

- *Žádost-Přístupu* (angl. Access-Request, kód: 1): Tato zpráva je generována RADIUS klientem vůči RADIUS serveru za účelem předání žadatelova požadavku.
- *Výzva-Přístupu* (angl. Access-Challenge, kód: 11): Tato zpráva je zaslána RADIUS serverem k RADIUS klientovi a obecně nese dotaz vůči žadateli k provedení určité akce.

- *Udělení-Přístupu* (angl. Access-Accept, kód: 2): Tato zpráva je zaslána RADIUS serverem vůči RADIUS klientovi oznamující úspěšné schválení požadavku.
- *Zamítnutí-Přístupu* (angl. Access-Reject, kód: 3): Tato zpráva zaslána RADIUS serverem značí zamítnutí požadavku žadatele.
- *Žádost-Účtování* (angl. Accounting-Request, kód: 4): Tuto zprávu zasílá RADIUS klient k RADIUS serveru pro předání účtovacích informací vztahujících se k službám poskytnutým žadateli.
- *Odpověď-Účtování* (angl. Accounting-Responce, kód: 5): Tato zpráva je odeslána RADIUS serverem k RADIUS klientovi pro potvrzení přijatých účtovacích informací a označení provedené účtovací funkce RADIUS serverem.
- *Stav-Klienta* (angl. Status-Server, kód: 12): Tato zpráva je experimentální a předmětem vývoje.
- *Stav-Serveru* (angl. Status-Client, kód: 13): Tato zpráva je taktéž experimentální a předmětem vývoje.
- *Rezervováno* (angl. Reserved, kód 255): Tato zpráva je rezervována dle specifikace.

Pro nové RADIUS specifikace, například z doporučení RFC 3576, již bylo definováno několik nových RADIUS zpráv (kódů). Z důvodu velkého nasazení základní specifikace protokolu RADIUS se RADIUS komunita snaží zajistit zpětnou kompatibilitu mezi nově zavedenými funkcemi a stávající specifikací. Mnoho z nových doporučení je kategorizováno jako informativní namísto standardních z důvodu možných problémů, které mohou způsobit vůči existující specifikaci.

Protokol RADIUS dokáže přenášet informace týkající se různých funkcí. Tyto informace jsou přenášeny formou takzvaných atributů. Každý z atributů může být samostatný balíček včetně informace o délce proměnné a strukturován dle formy typu, délky a hodnoty (TLV) viz tabulka 3.2.1. Užití nového datového typu TLV navýšilo množství přenášené informace a umožnilo atributy vnořit. Pro přenos atributů mezi RADIUS klientem a serverem obecně slouží hlavní tělo zprávy Žádost-Přístupu a Výzva-Přístupu. Atributy obsahují téměř veškeré informace potřebné k určení RADIUS operace, a tedy dvě různé zprávy Žádost-Přístupu nesoucí odlišné atributy provádí jiné funkce dle vlastních atributů. [2, 5]

3.2.1 Formát paketu protokolu RADIUS

Formát paketu protokolu RADIUS nemá složitou strukturu a skládá se z hlavičky, která obsahuje kód, identifikátor (ID), délku a autentizační pole a dále následované tělem paketu, které nese žádný či více atributů (tabulka 3.2.2). Konec seznamu atributů je určený polem délky v hlavičce RADIUS paketu.

Tabulka 3.2.1: RADIUS atribut dle TLV [2]

Typ atributu	Velikost do 1 oktetu (8 bitů tedy značí maximálně 255 možných atributů) k popisu možného typu atributu
Délka atributu	Délka atributu v oktetech, které značí celkovou délku těla atributu zahrnující pole typu a délky
Hodnota atributu	Obsahuje informace nesené atributem

Tabulka 3.2.2: Formát paketu RADIUS zprávy [2]

Název pole	**Podpoložka**	**Popis**
Hlavička	Kód	Určuje typ RADIUS paketu (např. Žádost-Přístupu)
	Identifikátor	Slouží k sjednocení žádosti a odpovědi
	Délka	Velikost 2 oktety k značení délky celé zprávy
	Autentizační pole	Velikost 16 oktetů, jehož obsah je vypočten dle vzorce
Atribut č. 1		První atribut v paketu
...		...
Atribut č. N		Poslední atribut v paketu

3.3 Rozšiřitelnost protokolu RADIUS

V kapitole 3.2 bylo řečeno, že většina podstatných informací je přenášena pomocí atributů uvnitř zpráv protokolu RADIUS. Toto znamená, že atributy poskytují možnost rozšíření funkčnosti RADIUS serveru pro interakci s mnoha dalšími entitami za různými účely. K podpoře scénářů specifických implementací je možné definovat takzvaný výrobcem-specifický atribut (VSA), který umožní různým výrobcům NAS interagovat s RADIUS serverem způsobem, který určí výrobce.

Základní specifikace protokolu RADIUS [6] specifikuje celkem 63 atributů, zatímco několik dalších atributů bylo definováno v pozdějších doporučeních. Samotný list atributů je tedy rozsáhlý a jejich rozpis by zde nebyl vhodný. Více informací o atributech je možné získat ze základní specifikace protokolu RADIUS [6] a kapitoly 3.1 a 4.5 z literatury [5]. Pracovní skupina z IETF se aktivně zabývá možností definovat nové atributy, které by rozšířily funkčnost protokolu RADIUS, ale zároveň zachovaly zpětnou kompatibilitu s existujícím nasazením protokolu RADIUS. Z důvodu omezení velikosti atributového pole typu na 8 bitů je celkový počet existujících atributů omezen na 255 možností. Teoreticky existuje možnost rozšířit funkčnost protokolu RADIUS způsobem vytvoření nových RADIUS zpráv, ale z důvodu zajištění zpětné kompatibility se pracovní skupina IETF zdržela takto rozhodnout. [2]

3.4 Spolehlivost přenosu protokolu RADIUS

RADIUS je možné považovat za protokol aplikační vrstvy, který přenáší data užitím transportního protokolu. Při návrhu doporučení protokolu RADIUS bylo určeno, že použití UDP je v rámci nasazení protokolu RADIUS vhodnější než TCP.

Důvodem tohoto rozhodnutí je například časově náročný proces ustanovení výměny zpráv, který zahrnuje držení stavů zařízení při použití TCP. Pro udržení stavu zařízení je nutné zajistit, aby existoval speciální kód či administrativní řešení pro klienty a servery za účelem zmírnění efektu ztráty napájení, restartování a vysokého síťového provozu. Tyto nároky na klienty a servery nejsou přítomné při volbě UDP, které umožňuje zajistit výměnu zpráv a držet ji otevřenou po celou dobu transakce. [2, 5]

3.5 Zabezpečení protokolu RADIUS

Bezpečnostní opatření jsou v systému RADIUS vcelku primitivní. K dispozici jsou dvě hlavní funkce, první je skrytí atributu (především hesla) a druhá je autentizace určitých zpráv. Obě funkce jsou provedeny s použitím typu hašovací funkce takzvané strávení-zprávy verze 5 (MD5) na heslo, které je sdíleno mezi RADIUS serverem a RADIUS klientem (NAS). Toto heslo se obecně nazývá sdílené heslo. [2]

3.5.1 Bezpečnostní nedostatky protokolu RADIUS

V případě autentizace zprávy či skrytí atributu protokol RADIUS umožňuje pouze jednu metodu kryptografického hesla, kterým je sdílené heslo mezi NAS a RADIUS serverem. Užití sdílených hesel jako základ pro poskytování bezpečnostních funkcí v rámci protokolu RADIUS způsobuje mnoho zranitelností při nasazení. Mezi některé z významných zranitelností patří:

- *Statická konfigurace sdílených hesel*: Protokol RADIUS nemá definovanou metodu pro dynamické a automatické sdílení hesel ve své základní specifikaci. Předem sdílená hesla typicky ručně konfiguruje NAS. Z důvodu velkého počtu zapojených NAS v mnoha sítích došlo k situaci, kdy síťový administrátor užívá shodné heslo pro všechny NAS za účelem zmírnění administrativní zátěže. Platnost sdílených hesel je většinou dlouhodobá a definice metody obnovení hesel nejsou specifikovány.

- *Dohledání sdíleného hesla*: Pro zabránění podvodného útoku používá RADIUS server zdrojovou adresu internetového protokolu (IP) uvnitř UDP paketu (namísto adresy IP NAS či ID atributů) k dohledání sdíleného hesla. Tento způsob vychází z potřeby podporovat zabezpečení skok-na-skok (angl. hop-by-hop) při návrhu RADIUS zástupců a částečně z důvodu, že atribut NAS-ID je přidán do těla zprávy Žádost-Přístupu. Toto ustanovení může potencionálně způsobit množství problémů v případech, kdy se IP adresa NAS změní. NAS může získat IP adresu i dynamicky prostřednictvím protokolu konfigurace hostitelského počítače (DHCP), jak tomu může být u mnoha aktivních bodů WLAN. Správa aktivních

bodů WLAN s velkým počtem přístupových bodů bez využití DHCP způsobuje značnou administrativní zátěž.

- *Řetězení zástupců*: Při nasazení RADIUS zástupců mezi NAS a RADIUS server dochází k procesu, kdy zařízení NAS sdílení hesla pouze s prvním AAA zástupcem namísto s koncovým RADIUS serverem, který je skutečný cílový adresát. Toto znamená, že důvěra mezi NAS a RADIUS serverem je pouze přechodná, tedy komunikace mezi NAS a RADIUS serverem je založena na řetězci důvěry namísto důvěry přímé. Je možné, že nastanou problémy zabezpečení a vznik podvodu v případě, kdy útočník jedná jako RADIUS zástupce.

- *Ochrana přenosu*: Metoda ukrytí atributu poskytuje selektivní ochranu aplikační vrstvy. V tomto případě není zajištěno zabezpečení ve formě autentizace či šifrování pro RADIUS zprávy, které jsou přenášeny na nižších vrstvách (UDP a IP). Toto v důsledku znamená náchylnost na podvodné útoky na IP adresu či změnu jiných atributů. [2]

3.6 Interakce protokolu RADIUS s EAP

Princip funkce rozšířeného autentizačního protokolu byl již popsán v kapitole 2.3. Pro zopakování zmíním, že EAP byl navržen pro podporu obecného autentizačního protokolu, jehož funkční rozšíření nemusí vyžadovat aktualizaci systému NAS. Protokol RADIUS v základní specifikaci umožňoval podporu pouze starších autentizačních protokolů jako PAP a CHAP. Protokol RADIUS byl rozšířen o podporu EAP z důvodu, že koncový server musí porozumět zprávám EAP a na základě obsahu zpráv poté provést autentizaci. Flexibilita a rozšiřitelnost protokolu RADIUS, zajištěná přenosem informací formou atributů, umožnila přenášet zprávou protokolu RADIUS i další zprávy jiných protokolů. EAP-RADIUS rámec umožňuje zapouzdření zpráv pro autentizační schéma EAP vložených do atributů, které jsou přenášeny v těle zprávy protokolu RADIUS.

Pojmem zapouzdření zpráv EAP je myšleno zahrnutí specifického RADIUS atributu EAP-Zpráva uvnitř těla RADIUS zprávy. Atributy těchto typů jsou specifikovány v doporučení RFC 3579 [7] týkající se podpory EAP pro protokol RADIUS. Toto doporučení konkrétně specifikuje podporu následujících atributů:

- *EAP-Zpráva atribut (typ 79)*: Tento atribut zapouzdřuje jeden fragment zprávy EAP, který obsahuje typ PPP, ID-požadavku, délku a pole EAP-typ.

- *Atribut autentizátoru zprávy (typ 80)*: Tento atribut zajišťuje integritu zprávy. [2]

3.7 Účtování v protokolu RADIUS

Základní specifikace protokolu RADIUS (RFC 2865 [6]) nedefinuje podporu pro účtovací procedury. Účtování protokolu RADIUS bylo dodatečně definováno v doporučení RFC 2866 [8].

Účtovací procedura je taktéž založena na bázi klient-server, kde RADIUS klient (NAS) předává účtovací informace od žadatele směrem k RADIUS serveru.

Účtovací procedury se skládají z dvou typů zpráv: Žádost-Účtování a Odpověď-Účtování, kde jsou obě zprávy přenášeny pomocí UDP. Zpráva Odpověď-Účtování je vždy generována RADIUS serverem na základy přijetí zprávy Žádost-Účtování, kterou zasílá NAS. Doporučení pro účtování v protokolu RADIUS [8] definuje sadu nových atributů k využití pro účtovací procedury. [2]

3.8 Zabezpečení a spolehlivost účtování v protokolu RADIUS

Z důvodu přenosu zpráv protokolu RADIUS přes UDP není zaručena celistvost přijatých dat a je tedy možné pozorovat ztrátu účtovacích paketů při síťovém provozu. Například při ztrátě zprávy Účtování-Stop můžou nastat problémy typu:

- Neúplnost či nesprávnost informací měřených služeb způsobí ztrátu na příjmech či rozpory se zákazníkem, které není možné vyřešit.
- Někteří poskytovatelé služeb mohou implementovat omezení počtu současně aktivních spojení, které uživatel může navázat. Při ztrátě účtovacích paketů, značících ukončení spoje, může být uživateli odepřena žádost o nové spojení.

Doporučení pro účtování v protokolu RADIUS [8] značí, aby klient pokračoval v odesílání zpráv Žádost-Účtování, dokud nebude doručeno potvrzení. K omezení přetížení je vhodné prodloužit časový limit a snížit počet přenosů na opakované zaslání žádosti. Pokud jde o zabezpečení účtování, tak požadavky i odpovědi jsou autentizovány pomocí hašovací funkce typu MD5 a sdíleného hesla mezi klientem a serverem. [2]

3.9 Podpora pro roaming a mobilitu v protokolu RADIUS

RADIUS nabízí velmi omezenou podporu pro mobilitu a vícedoménové operace. Většina podpory mobility uživatelů vychází z práce, kterou provedla pracovní skupina z IETF nazývaná „Roamingové operace“ [9].

Pojem roaming obvykle značí, když uživatel, který je běžně spojený s domácím poskytovatelem, vyžaduje z jiné lokality poskytnutí služby od sítě, kterou provozuje a vlastní jiný poskytovatel. Roamingové vztahy jsou tedy založené na smlouvách mezi dvěma a více poskytovateli. Podpora roamingových služeb v protokolu RADIUS je popsána v doporučení RFC 2607 [10], které definuje procedury pro řetězení zástupců. V diskuzi AAA je řetězení zástupců možné definovat jako procedury potřebné k předání paketů AAA mezi zařízením NAS a domácím RADIUS serverem užitím série zástupců, když se uživatel pohybuje v cizí doméně.

Doporučení RFC 2607 [10] pro protokol RADIUS definuje RADIUS zástupce jako uzel, který může být použit pro poskytování směrování autentizace a účtovacích zpráv mezi NAS a RADIUS serverem. Zástupce působí jako RADIUS server při práci s NAS, ale jedná jako RADIUS klient

při komunikaci s RADIUS serverem. Část doporučení se zabývá bezpečnostní zranitelností řetězení zástupců například z hlediska absence procedur pro směrování, nemožnosti skrytí atributů a náchylnosti na útok typu MITM. [2]

3.10 Problémy protokolu RADIUS

Při návrhu síťové architektury je věnována zvýšená pozornost integrovanému přístupu, který zohledňuje například mobilitu, technologie kvality služeb (QoS), kontrolu a zabezpečení přenosů. Role AAA serverů se stává více centrální a dále existuje vyšší nárok na interakci mezi AAA servery a dalšími subjekty v síti s požadavkem na AAA protokoly k udržení spolehlivosti, bezpečnosti a mobility.

V protokolu RADIUS se vyskytuje mnoho problémů z hlediska bezpečnosti a spolehlivosti. Protokol RADIUS dále neumožňuje podporu mobility v IP. Počet možných typů atributů je omezen na 255 společně s omezením délky atributové hodnoty pole. Tento fakt, dohromady s dalšími nedostatky sady RADIUS zpráv, omezuje využitelnost protokolu RADIUS do budoucna.

Protokol Diameter, jakožto nástupce protokolu RADIUS, překonává mnoho ze zmíněných nedostatků. Z tohoto důvodu se IETF již rozhodla uzavřít práci standardizace protokolu RADIUS. Protokol RADIUS je nicméně široce nasazen, a proto IETF určila pracovní skupinu „RADIUS rozšíření“ [11] k práci na standardizaci nových RADIUS atributů, které rozšiřují funkčnost protokolu RADIUS. Dále je kladen silný důraz na zachování zpětné kompatibility s existujícím nasazením protokolu RADIUS. Kompatibilita mezi protokoly RADIUS a Diameter je taktéž vážně zohledňována. [2]

4 Protokol Diameter

Přibližně ve stejnou dobu, kdy se pracovní skupina z IETF rozhodla o ukončení práce na protokolu RADIUS (začátek roku 2000), došlo k vzniku nové skupiny s názvem „pracovní skupina AAA“, která započala úkol nalezení nového nástupce stávajícího protokolu RADIUS. Skupina se rozhodla provést důkladné srovnání návrhů protokolů a za tímto účelem bylo vytvořeno nové doporučení RFC 2989 [12], které definuje kompletní soubor požadavků na AAA protokol. Mezi příklady definovaných požadavků patří:

- Škálovatelnost, převzetí služeb při selhání (angl. fail-over), vzájemná autentizace mezi klientem a serverem, zabezpečení na úrovni přenosu, důvěryhodnost a integrita datových objektů, přenos certifikátů, spolehlivý přenos AAA mechanismů, schopnost využití internetového protokolu čtvrté verze (IPv4) a internetového protokolu šesté verze (IPv6), podporu pro zástupce, podporu pro zprostředkovatele směrování a dále schopnost přenosu atributů specifických pro službu.

Pracovní skupina AAA následně požádala zástupce kandidujících protokolů k předložení návrhů, aby mohlo být zjištěno, zdali jsou požadavky z doporučení RFC 2989 splněny. Za hlavní kandidáty byly označeny tyto protokoly: jednoduchý protokol správy sítě (SNMP), protokol RADIUS++, protokol společné otevřené politiky služeb (COPS) a protokol Diameter.

Expertní tým provedl studii kandidujících protokolů a výsledky byly zveřejněny v doporučení RFC 3127 [13]. Protokol Diameter nejblíže splňoval požadavky AAA protokolu z doporučení RFC 2989 [12] a byl prohlášen za vítěze provedené studie. Pracovní skupina AAA se tedy rozhodla soustředit své úsilí na další rozšiřování a standardizaci protokolu Diameter. [2, 14]

4.1 Základní specifikace protokolu Diameter

Doporučení RFC 3588 [15], které bylo v roce 2012 nahrazeno doporučením RFC 6733 [16], definuje většinu stavebních prvků protokolu Diameter, jako je základní soubor zpráv, sadu atributů a jejich strukturu. Základní specifikace protokolu Diameter (na rozdíl od protokolu RADIUS) taktéž podrobně definuje řadu mezidoménových operací.

V této specifikaci byla nově definována takzvaná koncepce aplikací. Aplikace protokolu Diameter jsou služby, protokoly a procedury, které využívají prostředky Diameter serverů, zástupců a samotného protokolu Diameter. Všechny aplikace protokolu Diameter musí podporovat funkčnost definovanou v základní specifikaci protokolu Diameter.

Protokol RADIUS se zde velmi liší od způsobu, jakým protokol Diameter nahlíží na zařízení NAS. Komunikace s NAS za účelem procesu autentizace a autorizace je považována za aplikaci protokolu Diameter. Přes veškerý objem informací a pocit úplnosti nejsou v základní specifikaci protokolu Diameter zmíněny jakékoli podrobnosti o autentizačních a autorizačních procedurách, nebo dokonce interakce s NAS. Interakce protokolu Diameter s NAS za účelem autentizace je definovaná v jiném samostatném doporučení. Dále je v základní specifikaci předpokládáno,

že zprávy obsahující žádost o autorizaci jsou aplikačně specifické a jejich definice náleží jiným dokumentům. V specifikaci je ale definováno, že Diameter klient (NAS) je entitou, která zajišťuje vydání autentizačních či autorizačních požadavků směrem od žadatele vůči Diameter serveru.

V kontrastu s protokolem RADIUS jsou v základní specifikaci protokolu Diameter definovány metody účtování a jejich struktura zpráv. Důvodem může být, že podpora účtovacích procedur je podmíněna pro všechny aplikace protokolu Diameter. [2]

4.2 Přenosový profil protokolu Diameter

V kapitole 3.4 byly zmíněny nedostatky spolehlivosti přenosu protokolu RADIUS z důvodu využití transportního protokolu typu UDP. Základní specifikace protokolu Diameter definuje transportní spojení mezi Diameter uzly (angl. nodes) užitím TCP či SCTP připojení. Dále je specifikací vyžadováno, aby Diameter agenti a server podporovali souběžně TCP a SCTP, zatímco klient musí podporovat alespoň jeden z protokolů.

Na rozdíl od protokolu RADIUS je podstatné, že protokol Diameter vyžaduje k chodu použití spolehlivého transportního protokolu. [2]

4.3 Aplikace protokolu Diameter

Základní specifikace protokolu Diameter popisuje pouze podporu pro účtovací metody, zatímco jiné protokoly využívající Diameter protokol či servery jsou považovány za aplikace. Tyto aplikace jsou popsány v samostatných dokumentech specifických pro danou aplikaci.

Lze bezpečně předpokládat, že ne každý uzel protokolu Diameter nasazený v infrastruktuře bude podporovat všechny dostupné aplikace. V případě vzájemné spolupráce dvou uzlů jménem některé aplikace je nutné nejprve ověřit společnou podporu pro danou aplikaci. V protokolu Diameter je ověření podpory aplikace poskytováno funkcí s názvem vyjednávání schopností, která určí možnost podpory na základě několika zpráv obsahující identifikátory aplikací. Mezi některé z aplikací protokolu Diameter se specifickým ID patří například účtování, Diameter NAS, mobilita v IP či Diameter EAP. [2]

4.4 Typy uzlů protokolu Diameter a jejich role

V rámci infrastruktury protokolu Diameter se vyskytuje několik nových entit, které se liší od entit protokolu RADIUS, jenž byly popsány v kapitole 3.1. Pro zajištění spolehlivého, bezpečného a směrovaného chování jsou funkce protokolu Diameter pevně specifikovány. Základní specifikace protokolu Diameter kategorizuje zúčastněné entity na základě jejich role při přenosu zpráv protokolu Diameter:

- *Diameter uzel*: Uzel, který má přímé přenosové spojení s jiným uzlem.

- *Diameter klient*: Zařízení na okraji sítě, které provádí řízení přístupu. Může se jednat o NAS nebo cizí agenta mobility v IP. Koncový uživatel, který si žádá o přístup do sítě, není klient, protože uživatel se nepodílí na signalizaci protokolu Diameter.

- *Diameter server*: Server je zařízení, které zpracovává AAA požadavky pro určitou oblast. Server musí podporovat základní specifikaci protokolu Diameter a dalších aplikací užívaných v oblasti.

- *Diameter agent*: Agent je typ uzlu zajišťující služby zprostředkování (angl. relay), zastupování (zástupce), přesměrování a překládání informací. Zprostředkovatelé neprovádí vlastní rozhodnutí dle politiky, ale přenáší zprávy založené na atributech a speciálních směrovacích tabulkách. Zástupci již sledují různé stavy v zařízení NAS a mohou generovat zprávy Zamítnutí-Přístupu při porušení politiky. Agenti přesměrování slouží k odkázání klienta k serveru předáním zpráv na základě vlastní konfigurace. Překladatelé provádí překlady mezi protokolem Diameter a jiným protokolem z rodiny AAA, například protokolem RADIUS. Překladatelé byli speciálně navrženi k zajištění zpětné kompatibility. [2]

4.5 Zprávy protokolu Diameter

Zprávy protokolu Diameter jsou využívány k přenosu různých aplikací či AAA informací stejným způsobem, jako je tomu i protokolu RADIUS. Informace nesena Diameter zprávou se typicky označuje atribut, který je formátován jako pár atribut-hodnota (AVP). Atributy dle formátu AVP existují taktéž v protokolu RADIUS.

V kapitole 3.1 bylo řečeno, že protokol RADIUS funguje na bázi server-klient, kdy jsou požadavky vždy vydávány klientem, ale odpovědi (výzvy nebo přijetí/zamítnutí) jsou vytvářeny serverem. Protokol Diameter se architekturou odlišuje užitím báze rovnocenných účastníků (angl. peer-to-peer), což znamená, že klient i server mohou vytvářet žádosti i odpovědi libovolně. Namísto terminologie různých typů zpráv protokol Diameter využívá koncept příkazů. Příkazy se od sebe odlišují pomocí příkazového kódu, který specifikuje typ funkce, jenž má v úmyslu provést Diameter zpráva. [2]

4.5.1 Formát zprávy protokolu Diameter

Dle tabulky 4.5.1 je Diameter zpráva složena ze standardní hlavičky a počtu jednoho či více AVP. Hlavička je v tabulce 4.5.1 označena v šedé barvě a obsahuje několik parametrů, mezi které patří:

- *Verze*: Značí verzi použitého protokolu Diameter.

- *Délka zprávy*: Indikuje vymezenou délku celé Diameter zprávy v jednotkách oktetů.

- *Příkazové příznaky*: Je pole specifikující celkem čtyři příznaky:

  - *R příznak* (Žádost): Určuje, zdali je zpráva žádost či odpověď.
  - *P příznak* (Zástupce): Značí, je-li možné zprávu zprostředkovat, přesměrovat, předat, nebo je nutné zprávu lokálně zpracovat.
  - *E příznak* (Chyba): Slouží pro informování o protokolových a sémantických chybách ve zprávě.
  - *T příznak*: Určuje, že zpráva může být opakovaně zaslána v případě selhání přenosu či k odstranění duplicitních zpráv.

- *Příkazový kód*: Označuje příkaz přidružený k zprávě, jako například "přerušení žádosti o relaci" nebo "účtovací odpověď". Každá Diameter zpráva musí obsahovat příkazový kód, aby bylo možné vyhodnotit, jakou akci je nutné provést.
- *ID-aplikace*: Značí konkrétní aplikaci, pro kterou je zpráva určena (např. mobilita v IP)
- *Skok-na-skok (angl. hop-by-hop) identifikátor*: Používá se k nalezení shody žádosti s odpovědí v daném skoku. Odesílatel žádosti musí zajistit jedinečnost identifikátoru pro dané spojení.
- *Konec-k-konci (angl. end-to-end) identifikátor*: Je využíván k detekci duplicitních zpráv na základě unikátního kódu. [2, 16]

Tabulka 4.5.1: Formát zprávy protokolu Diameter [16]

Bit: 0 1 2 3 4 5 6 7	**0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7**
Verze	Délka zprávy
Příkazové příznaky	Příkazový kód
ID-aplikace	
Skok-na-skok identifikátor	
Konec-k-konci identifikátor	
AVP data ...	

4.6 Atributový formát páru atribut-hodnota

V obsahu kapitoly 4.5 byly zmíněny AVP, které přenáší většinu informací uvnitř Diameter zpráv. AVP je možné libovolně přidávat ke zprávám, pokud jsou minimální požadované AVP ve zprávě taktéž zahrnuty. Formát AVP je uveden v tabulce 4.6.1, kde je hlavička AVP vyznačena v šedé barvě a mezi obsažené parametry patří:

- *AVP kód*: Identifikuje typ informací (atributů) obsažených v atributových datech. AVP kód obsahuje hodnoty, které jsou standardizovány IETF. Nové aplikace by měly využít již existujících AVP v maximální možné míře. V protokolu Diameter jsou kódy v rozmezí 0-255 vyhrazené pro zajištění zpětné kompatibility s protokolem RADIUS.

- *V příznak*: Označuje přítomnost nepovinného pole ID-výrobce v hlavičce AVP.
- *M příznak*: Je povinný a značí, zda Diameter uzel vyžaduje, aby jeho účastník podporovat daný atribut k zpracování zprávy.
- *P příznak*: Vyznačuje potřebu šifrování pro zabezpečení mezi koncovými body.
- *R příznak*: Značí existenci celkem pěti rezervovaných příznaků.
- *AVP délka*: Je vyhrazený třemi oktety a hodnota parametru označuje počet oktetů celého AVP.
- *ID-výrobce*: Je nepovinný parametr obsahující informace od výrobce, které musí podléhat procesu standardizace.
- *AVP data*: Obsahuje žádný či více oktetů informací specifických k atributu. Formát a délka dat informací odpovídá hodnotám v parametrech *AVP kód* a *AVP délka*. [2, 16]

Tabulka 4.6.1: AVP formát Diameter atributu [16]

Bit: 0 1 2 3 4 5 6 7	**0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7 0 1 2 3 4 5 6 7**
AVP kód	
VMPRRRR	AVP délka
ID-výrobce (nepovinné)	
AVP data ...	

4.7 Směrovací koncept protokolu Diameter

Základní specifikace protokolu RADIUS neposkytuje jasné pokyny o provozu RADIUS zástupců v prostředí s více doménami a v roamingových scénářích. Proces směrování RADIUS paketů není vždy jasný z RADIUS specifikací. Protokol Diameter již zřetelně definuje jasné směrovací postupy na Diameter uzlech.

Tabulka účastníků (angl. peer table) protokolu Diameter je využívána k předávání informací tabulce směrovací oblasti, která se na ni odkazuje. Každý Diameter uzel udržuje tabulku účastníků obsahující záznamy o totožnosti, užitečných informacích o stavu a časově závislé údaje. Tabulku směrovací oblasti využívají Diameter agenti pro předání zpráv k dalšímu cíli, který může být umístěn v jiné oblasti. [2]

4.8 Výměna zpráv Diameter server-NAS

Příklad výměny zpráv mezi Diameter serverem a zařízením NAS je znázorněný na obrázku 4.8.1. Výměna zpráv mezi žadatelem a NAS není považována za výměnu Diameter zpráv, ale je zahrnuta pouze pro úplnost. Proces výměny zpráv se provádí následovně:

- Při přijetí žádosti přístupu k službě či autentizaci od žadatele dojde k vytvoření zprávy AA-Žádost zařízením NAS, které obsahuje identitu a autentizační informace od žadatele. NAS poté zašle zprávu vůči Diameter serveru.

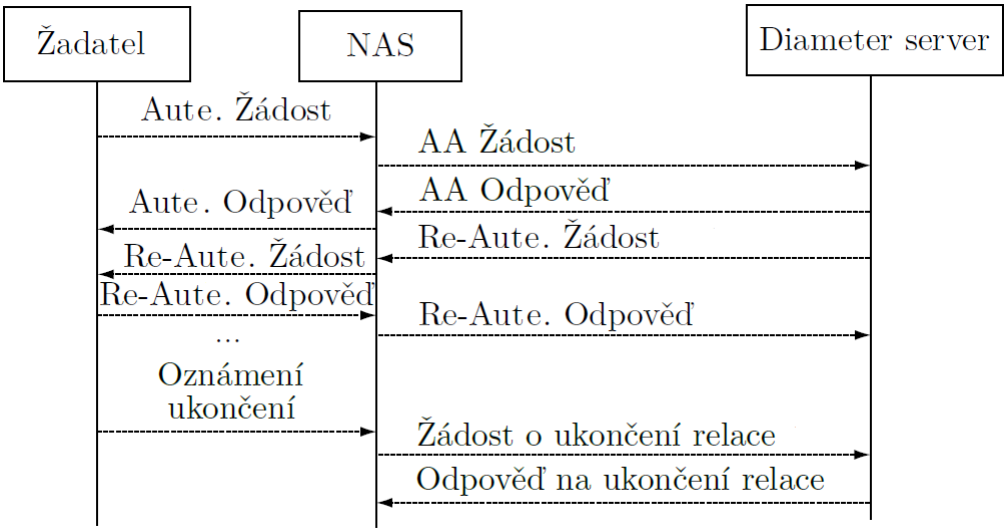
- Diameter server zpracuje požadavek a vrátí zprávu AA-Odpověď, včetně informace o autorizaci nebo AVP *Výsledek-Kód*, který značí určité selhání. Server může taktéž naznačit, že je vyžadována další výměna zpráv pro dokončení procesu autentizace. Protokol Diameter taktéž umožňuje přijímat samotné autorizační žádosti, které neobsahují data související s autentizací. Užití tohoto typu autorizační žádosti může vést k bezpečnostnímu riziku v případě chybné implementace. Dále pak neexistují odpovídající funkce v protokolu RADIUS a při překladu mezi protokoly by mohlo dojít k selhání.

- Při úspěšné autentizaci a autorizaci dojde ke spuštění kontextu relace zařízením NAS. V případě implementovaného účtování dojde taktéž k zaslání zprávy pro zaznamenání nové relace. Diameter server informuje NAS o maximální povolené době před opětovnou autentizací a autorizací užitím AVP *Autorizace-Životnost*.

- Při spuštěné relaci může Diameter server vydávat nevyžádané žádosti o opětovnou autorizaci a autentizace před uplynutím doby autorizace, aby bylo možné zabránit ukončení služby.

- Žadatel oznamuje zařízení NAS, že vyžaduje ukončit existující relaci. Zařízení NAS vydává žádost o ukončení relace vůči Diameter serveru. V případě ukončení relace účtování je server taktéž obeznámen.

- Po přijetí a zpracování platné žádosti o ukončení relace dojde k ukončení relace ze strany Diameter serveru a je zaslána odpověď včetně AVP *Výsledek-Kód*. Při ukončení relace Diameter server uvolní všechny prostředky, které jsou svázány s ID pro ukončenou relaci. Taktéž dochází k uvolnění všech Diameter zástupců, kteří se podíleli na relaci. [2]



Obrázek 4.8.1: Výměna zpráv Diameter server-NAS [2]

4.9 Výhody užití protokolu Diameter vůči RADIUS

Součástí této kapitoly je uvedení řady vylepšení, které protokol Diameter nabízí oproti protokolu RADIUS.

4.9.1 Převzetí služeb při selhání

Převzetí služeb při selhání je definováno jako proces předání všech nevyřízených požadavků směrované určitému agentovi vůči některému z jiných agentů v případě, je-li zjištěna porucha přenosu. Protokol RADIUS nedefinuje standardní mechanismus selhání a různé RADIUS implementace se tedy mohou vzájemně odlišovat. [2]

Protokol Diameter naopak nabízí vyšší odolnost vůči poruchám přenosu a poskytuje dobře definované chování při selhání. Mezi definované chování patří potvrzování na aplikační vrstvě, specifické hlídací (angl. watchdog) mechanismy k detekci nulové aktivity a fronta zpráv pro každého z účastníků. [2]

4.9.2 Zprávy iniciované serverem

Podpora zpráv iniciovaných serverem je v protokolu RADIUS pouze volitelná a je tedy obtížné implementovat funkce jako nevyžádané odpojení nebo opakovaná autentizace a autorizace na vyžádání. Podpora pro serverem iniciované zprávy je povinná v protokolu Diameter. [2]

4.9.3 Spolehlivý přenos

Užití transportního protokolu UDP a nedostupnost opakovaného přenosu způsobuje v protokolu RADIUS problémy ve spolehlivosti přenosu dat zejména u služby účtování, kde se ztráta paketů

může přenést do ztráty na příjmech. Protokol Diameter již umožňuje využít spolehlivé přenosové mechanismy TCP a SCTP. [2]

4.9.4 Vyjednávání schopností

Protokol RADIUS neumožňuje klientovi a serveru vzájemné doložení podpory pro různé atributy a dále nepodporuje chybové zprávy. To znamená, že objevování a vyjednávání služeb může být velmi obtížné při použití protokolu RADIUS. Zatímco protokol Diameter již nabízí podporu pro zpracování chyb, vyjednávání schopností a způsob ověření podpory pro AVP (užitím povinných příznaků ve zprávě). [2]

4.9.5 Problémy zabezpečení a slyšitelnosti

Mnoho z bezpečnostních nedostatků protokolu RADIUS bylo zmíněno v kapitole 3.5.1. Mezi některé z nedostatků patří statická konfigurace sdílených hesel a jejich dohledání, řetězení zástupců, nedostatečná ochrana přenosu. Protokol Diameter definuje bezpečnost na přenosové vrstvě a zabezpečení koncových bodů způsobem, že vyžaduje povinnou podporu zabezpečení internetového protokolu (IPSec) pro různé aplikace a volitelnou podporu TLS mezi účastníky. [2]

4.9.6 Podpora pro agenty a mezidoménový roaming

Protokol RADIUS neposkytuje jasnou podporu agentům a zástupcům. Z důvodu chybějící podpory řetězení zástupců vzniká u protokolu RADIUS zranitelnost vůči útokům a podvodům při využívání roamingových služeb. Protokol Diameter již zřetelně definuje roli a chování agentů či zástupců poskytnutím podpory mezidoménového roamingu, směrování zpráv a zabezpečení přenosové vrstvy. [2]

4.9.7 Nalezení účastníků v síti

Implementace protokolu RADIUS typicky vyžaduje manuální konfiguraci názvů hostitelů či IP adres serverů/klientů a sdílených hesel. V důsledku poté vzniká velká administrativní zátěž a pokušení opakovaného použití sdíleného hesla u více klientů (NAS). Protokol Diameter již umožňuje dynamické objevování účastníků užitím systému doménových jmen (DNS). [2]

4.9.8 Zpětná kompatibilita s protokolem RADIUS

Zatímco protokoly RADIUS a Diameter nesdílí společný formát zpráv, bylo vynaloženo značné úsilí k zajištění zpětné kompatibility mezi oběma protokoly v případě nasazení ve společné síti. Je očekáváno, že překlady budou muset probíhat přes určité brány umožňující komunikaci mezi staršími RADIUS zařízeními a Diameter agenty. [2]

4.10 Interakce protokolů Diameter a RADIUS

Je možné, že protokoly RADIUS a Diameter budou koexistovat v rámci jedné administrativy po dlouhou migrační dobu. V procesu návrhu protokolu Diameter byla zahrnuta i možnost případné koexistence obou protokolů. Příkladem je, že prostor RADIUS atributů je zahrnutý uvnitř protokolu Diameter, aby se eliminovala potřeba případné konverze.

Diameter aplikace NAS se zaměřuje na procesy autentizace a autorizace, a proto sdílí nejvíce podobností s protokolem RADIUS. Z tohoto důvodu popisuje specifikace o NAS aplikaci interoperabilitu mezi oběma protokoly a dále způsob implementace. Tato interoperabilita je představena architekturou skládající se z odlišných RADIUS a Diameter systémů, které komunikují s pomocí překladatelského agenta. Jelikož existují rozsáhle rozdíly mezi funkcemi obou protokolů, může existovat mnoho variant a implementací překladatelského agenta, který nemusí být standardizován IETF.

Dále Diameter aplikace NAS popisuje mnoho požadavků a procedur konverzí pro překladatelské agenty RADIUS-Diameter. V navazujícím seznamu nejsou zmíněny podrobnosti o překladu zpráv, atributů, AVP či služeb, ale jsou zahrnuty určité představy o problémech překladů zpráv mezi protokoly RADIUS a Diameter:

- Bezpečnostní mechanismy protokolu RADIUS jsou typu skok-na-skok, zatímco protokol Diameter může využít mechanismu konec-ke-konci. Diameter agent bude muset dešifrovat RADIUS zprávy, atributy a další informace způsobem specifickým pro protokol Diameter. Například, když překladatelský agent obdrží RADIUS zprávu obsahující atribut Uživatel-Heslo šifrovaný sdíleným heslem, musí agent dešifrovat heslo a předat informace o hesle uvnitř zprávy protokolu Diameter, která je chráněna jinými mechanismy zabezpečení. Hodnotu autentizátoru v RADIUS zprávě musí ověřit překladatelský agent, ale nesmí být zahrnuta v Diameter zprávě vytvořené agentem.

- Protokol RADIUS nepodporuje bázi rovnocenných účastníků či serverem iniciované zprávy, zatímco protokol Diameter definuje velké množství příkazových kódů, které lze použít jak ve zprávách typů žádost, tak i odpověď. Při opakovaném vyjednávání může protokol RADIUS umožnit zaslání zprávy Žádost-Přístupu odeslanou klientem nebo zprávu Výzva-Přístupu odeslanou serverem. Překladatelský agent musí vytvořit zprávy založené na příkazech protokolu Diameter.

- RADIUS servery jsou považovány za bezstavové (nedrží informaci o aktuální relaci), zatímco Diameter uzly jiř udržují stav. Překladatelský agent tedy může mít zkreslený obraz o celkové relaci.

- Protokol Diameter definuje AVP ve formátu plně specifikovaného doménového jména (FQDN), zatímco atributy protokolu RADIUS jsou jiného formátu. Překladatelský agent musí zaměnit formát informace podle systému, do kterého je zpráva předávána. Hlavním

příkladem je převod RADIUS AVP typu NAS-IP-adresa do Diameter AVP typu Počátek-Hostitel ve formátu FQDN.

- Protokol Diameter podporuje takzvané seskupené AVP. Když překladatelský agent obdrží RADIUS atributy, které musí být součástí seskupeného AVP, musí tyto atributy sestavit do tvaru seskupených Diameter AVP. Příkladem je zacházení s atributem *CHAP-Heslo*. RADIUS atribut CHAP-Heslo obsahuje odpověď v atributovém poli data, ale parametr CHAP-ID je umístěn v záhlaví atributu. Protokol Diameter definuje seskupený AVP typu *CHAP-Auntentizace*, který obsahuje odpověď CHAP a *CHAP-ID* v seskupeném AVP. Konverzi musí tedy provést překladatelský agent.

Stručně řečeno, překladatelský agent působí jako brána odpovědná za interoperabilitu mezi protokoly RADIUS a Diameter. Specifikace protokolu Diameter ale neuvádí podrobnosti operace těchto agentů. Proto nelze předpokládat, že implementace samotné specifikace protokolu Diameter povede k připravenosti zpětné kompatibility mezi oběma protokoly. [2]

5 Prostředí pro praktickou část

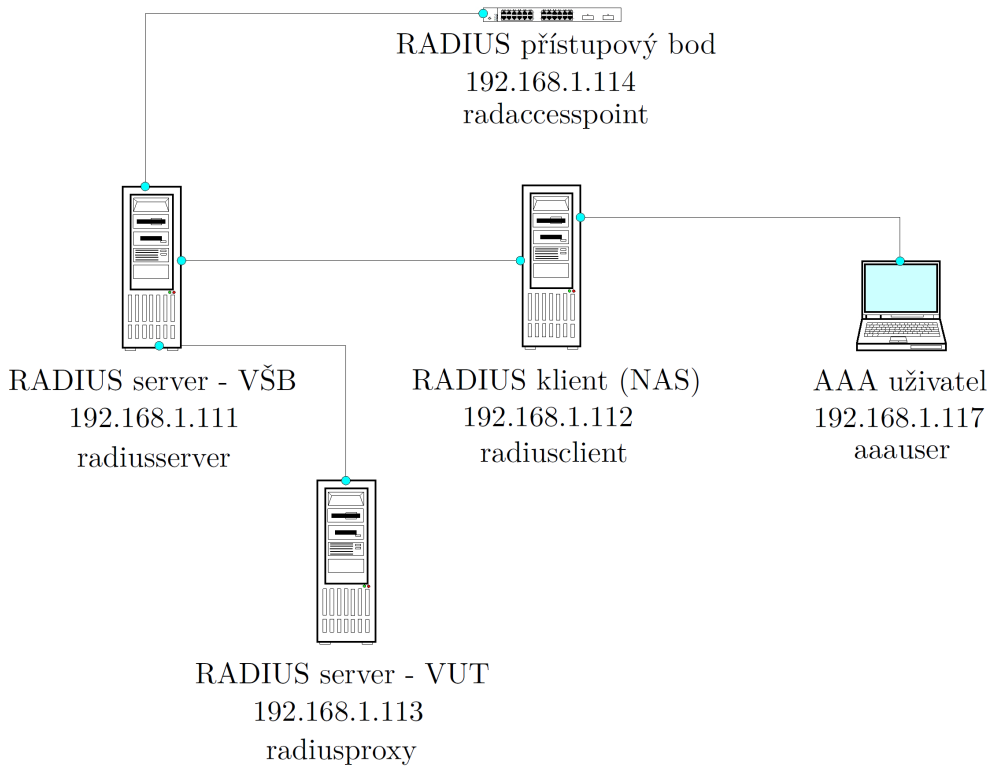
5.1 Virtualizační programové prostředí

Řešení a ladění návrhu praktické části diplomové práce jsem primárně provedl ve virtualizačním prostředí VMware Workstation 14 Pro (verze 14.0.0), které je ve zkušební verzi volně dostupné na internetových stránkách vývojáře programu [17].

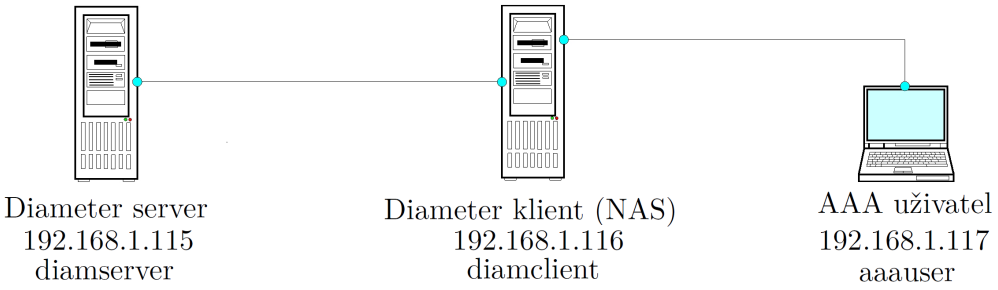
Programové prostředí VMware umožňuje na jednom fyzickém počítači spustit několik takzvaných virtuálních strojů. Virtuální stroj má vyhrazené fyzické prostředky hostitelského počítače a odděleně využívá vlastní operační systém. Pro konfiguraci praktické části jsem na virtuální stroje nainstaloval operační systém Ubuntu 14.04.5 LTS (Trusty Tahr), který je taktéž volně dostupný z internetových stránek vývojáře [18].

5.2 Síťové schéma virtuálních strojů

Pro návrh síťové topologie jsem využil celkem sedm unikátních virtuálních strojů se shodným operačním systémem. Síťové schéma konfigurace protokolu RADIUS je vyobrazené na obrázku 5.2.1 a schéma protokolu Diameter na obrázku 5.2.2.



Obrázek 5.2.1: Síťové schéma konfigurace protokolu RADIUS



Obrázek 5.2.2: Síťové schéma konfigurace protokolu Diameter

Síťové zapojení vychází z modelu autentizace tří stran, které bylo popsáno v kapitole 1.1.4.2. V síťovém schématu můžete vidět, že každé ze zařízení má přiřazenou jednoznačnou IP adresu a název hostitele. Přiřazené IP adresy jsou zde vyznačeny ve tvaru IPv4, ale implementace je plně funkční i pro IPv6 adresy.

5.3 Přiřazení IP adresy a názvu hostitele

Ve svém zapojení jsem zvolil přístup rezervování IP adres jednotlivým virtuálním strojům na základě jejich adresy přístupu k médiím (MAC). Přiřazení IP adresy na základě MAC adresy lze provést ve webovém rozhraní fyzického směrovače. Pro přiřazení IP adresy k názvu hostitele virtuálního stroje je nutné upravit soubor */etc/hosts*.

Je podstatné, aby každý z virtuálních strojů ze schématu zapojení měl ve svém souboru */etc/hosts* zahrnuté informace z výpisu 1.

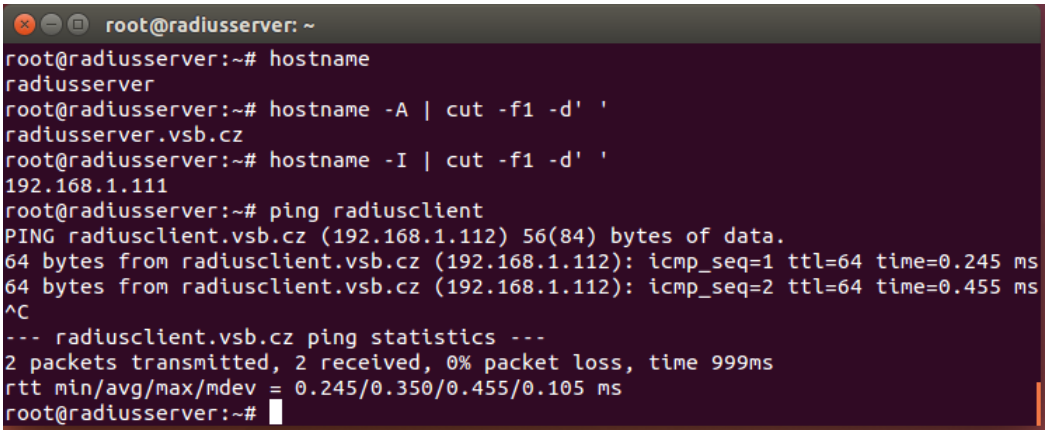
```
1 192.168.1.111    radiusserver.vsb.cz        radiusserver
2 192.168.1.112    radiusclient.vsb.cz        radiusclient
3 192.168.1.113    radiusproxy.vsb.cz         radiusproxy
4 192.168.1.114    radiusccesspoint.vsb.cz    radiusaccesspoint
5 192.168.1.115    diamserver.vsb.cz          diamserver
6 192.168.1.116    diamclient.vsb.cz          diamclient
7 192.168.1.117    aaauser.vsb.cz             aaauser
```

Výpis 1: Vzorový obsah souboru */etc/hosts*

Kromě IP adresy a názvu hostitele je definováno i FQDN, jenž pro zopakování značí plně specifikované doménové jméno. Definice FQDN na virtuálních strojích je nutná pro úspěšnou konfiguraci protokolu Diameter, která bude objasněna v pozdější kapitole.

Pro přihlášení pod nově definovaným doménovým jménem je nutné otevřít příkazový řádek (aplikace Terminal) a použít příkaz *hostname* (např. *hostname radiusserver*). Nejprve se musíte odhlásit z výchozího názvu hostitele příkazem *exit*. Na závěr použijete příkaz *sudo -i* a zadáte uživatelské heslo pro administrátora.

Důvodem definice IP adres, názvů hostitelů a FQDN před samotnou implementací je zaručit vyšší pružnost a přehlednost konfigurace. V konfiguračních souborech programů a služeb se poté na virtuální stroje můžeme odkazovat názvem hostitele bez nutnosti zasahovat do kódu při změně přiřazených IP adres. Pro ověření správně nastavených virtuálních strojů můžete použít příkazy dle obrázku 5.3.1.



Obrázek 5.3.1: Ověření nastavení virtuálních strojů před implementací

Každý z virtuálních strojů musí dostat úspěšnou odpověď na příkaz *ping*, jenž ověří možnost komunikace mezi dvěma virtuálními stroji. Provedením vstupní konfigurace této kapitoly jsou virtuální stroje připraveny pro nasazení implementace protokolů RADIUS a Diameter.

5.4 Skriptovací soubory

Unixové operační systémy umožňují spouštět takzvané skriptovací soubory užitím některého z interpretů příkazového řádku (např. Bash). Skriptovací soubor (někdy též shellový skript) je textový soubor, který obsahuje řadu příkazů, jenž interpret provede v sekvenčním pořadí. Účelem využití skriptovacích souborů je výrazně usnadnit konfiguraci a ladění jakékoliv implementace. Síťový administrátor tedy může na virtuální stroj nasadit kompletní implementaci programu či služby pouze spuštěním skriptovacího souboru.

Pro vlastní implementaci protokolů RADIUS a Diameter jsem vytvořil celkem osm různých skriptovacích souborů, které jsou volně dostupné k použití na přiloženém kompaktním disku (CD) v adresáři */skriptovaci_soubory/* a taktéž na internetových stránkách [19].

1. *hostname_setup.sh*: Slouží pro definování obsahu souboru */etc/hosts* dle vlastní potřeby. Po spuštění skriptovacího souboru se v totožném adresáři vytvoří nový skriptovací souboru s názvem *hostname_apply.sh*, který po spuštění přidá na závěr souboru */etc/hosts* nově definované údaje. Při implementaci tedy stačí jednou vygenerovat soubor *hostname_apply.sh* a přenést jej na ostatní virtuální stroje bez nutnosti manuálně upravovat soubor */etc/hosts*.

2. *freeradius_install_server.sh* (*radiusserver*): Po spuštění skriptovacího souboru se provede instalace a konfigurace hlavního RADIUS serveru pro VŠB doménu. Dále proběhne instalace a konfigurace databáze strukturovaného dotazovacího jazyka (SQL) typu MYSQL a databáze odlehčeného protokolu pro přístup k adresářům (LDAP). Zmíněné databáze slouží ke správě uživatelských údajů a účtovacích dat.

3. *freeradius_install_client.sh* (*radiusclient*): Pro instalaci a konfiguraci RADIUS klienta (NAS) stačí spustit tento skriptovací soubor. Na virtuální stroj se nainstaluje simulační program JRadius Simulator [20] s grafickým rozhraním a simulační nástroje *radtest* a *radclient*, které umožňují ověření funkčnosti nasazené implementace. Dále se nainstaluje a nakonfiguruje program Secure Shell (SSH) pro správu vzdáleného počítače, serverová část protokolu přenosu souborů (FTP), zásuvné autentizační moduly (PAM) a démon přístupového bodu hostitele (Hostapd).

4. *freeradius_install_proxy.sh* (*radiusproxy*): Druhý RADIUS server pro doménu VUT se zprovozní použitím tohoto skriptovacího souboru. Pro oba RADIUS servery je definována vazba na základě domény, z které přistupuje uživatel žádající o přístup. Jeden z RADIUS serverů se vždy chová jako zástupce (angl. proxy) a předává RADIUS zprávy vůči RADIUS serveru odpovídající domény.

5. *freeradius_install_access_point.sh* (*radiusaccesspoint*): Jedná se o velmi krátký skriptovací soubor, který nainstaluje pouze simulační nástroje *radtest* a *radclient*. Slouží pro ověření funkčnosti přístupového bodu, jenž je konfigurovaný na RADIUS serveru pro VŠB doménu.

6. *freediameter_install_server.sh* (*diamserver*): Tento skriptovací soubor nainstaluje a nakonfiguruje Diameter server. Taktéž se nainstaluje a naplní MYSQL databáze pro správu uživatelských údajů a vygenerují se SSL certifikáty společně s certifikační autoritou.

7. *freediameter_install_client.sh* (*diamclient*): Diameter klient (NAS) se nainstaluje a nakonfiguruje spuštěním tohoto skriptovacího souboru. Jedná se o velmi podobný skriptovací soubor jako je *freediameter_install_server.sh*, jelikož protokol Diameter považuje server a klient za rovnocenné účastníky (angl. peers). Rozdíl ve funkčnosti se obecně určuje nasazením určitých Diameter aplikací. Dále se opět nainstaluje program Hostapd, jakož tomu bylo u RADIUS klienta.

8. *aaa_install_wpa_supplicant.sh* (*aaauser*): AAA uživatel je v síťovém schématu zavedený pro úplné ověření funkčnosti nasazené implementace protokolů RADIUS a Diameter. Při spuštění skriptovacího souboru se nainstaluje a nakonfiguruje suplikant, jenž zajišťuje bezpečnou komunikaci v bezdrátových a pevných sítích v podobě démona s názvem wpa_supplicant. Program wpa_supplicant naváže spojení s programem Hostapd a zavede

komunikaci v případě, kdy je uživatel úspěšně autentizován a autorizován vůči RADIUS či Diameter serveru. Dále AAA uživatel slouží pro ověření funkčnosti SSH a FTP služeb.

Informace této kapitoly by měly sloužit převážně k nastínění obsahu praktické části práce. Hlubší zavedení do problematiky implementace protokolů RADIUS a Diameter bude předmětem navazujících kapitol. Dále doporučuji souběžně s textem práce pročítat i obsah skriptovacích souborů například v programu Notepad++, který je volně dostupný na internetových stránkách vývojáře [21].

Skriptovací soubory jsem strukturoval s určitou logickou návazností, která odpovídá posloupnosti textu práce. Na začátku každého skriptovacího souboru můžete nalézt sérii číslovaných příkazů, kterými lze lokálně i vzdáleně ověřit funkčnost nasazení implementace užitím příkazového řádku. V skriptovacích souborech jsou zahrnuté i textové komentáře pro orientaci v kódu a další rozvedení problematiky, která nemusí být plně vysvětlena v textu práce.

Pro přizpůsobení skriptovacích souborů dle konkrétního síťového schématu je možné upravit proměnné parametry, které jsou umístěné ihned za číslovanými příkazy pro simulaci. Na základě změny hodnot těchto proměnných parametrů je možné uzpůsobit implementaci z hlediska funkčnosti. Změnou proměnných parametrů lze například určit sdílené heslo mezi RADIUS serverem a RADIUS klientem, zvolit instalaci programu Hostapd, povolit konfiguraci určitých zásuvných autentizačních modulů či uzpůsobit konfiguraci pro IPv6. Obecnou strukturu proměnných parametrů ze skriptovacího souboru *freeradius_install_client.sh* můžete vidět ve výpisu 2.

```
1  SERVER_CLIENT_SHARED_PASSWORD="sharedpass123"
2  INSTALL_JRADIUS_SOFTWARE=false
3  INSTALL_HOSTAPD_DAEMON=true
4  INSTALL_HOSTAPD_INTERFACE_TYPE=eth0
5  INSTALL_PAM_SUPPORT=true
6  INSTALL_PAM_IPV4_ONLY=true
7  INSTALL_PAM_IPV4_IPV6_BOTH=false
8  OPERATING_SYSTEM_TYPE_i386_linux_gnu=true
9  OPERATING_SYSTEM_type_x86_64_linux_gnu=false
10 ENABLE_PAM_SSH=true
11 ENABLE_PAM_SUDO=true
12 ENABLE_PAM_FTP=true
```

Výpis 2: Část proměnných parametrů skriptovacího souboru *freeradius_install_client.sh*

6 Implementace protokolu RADIUS

6.1 Instalační program FreeRADIUS

Pro návrh implementace protokolu RADIUS jsem využil programu FreeRADIUS 2.2.10, jehož zdrojový kód je volně dostupný na stránkách projektu FreeRADIUS [22]. FreeRADIUS je modulární, vysoce výkonná implementace protokolu RADIUS s všeobecnou veřejnou licencí, verze 2 (GPLv2) projektu GNU. FreeRADIUS nabízí podporu většiny obecných autentizačních a databázových protokolů a mnohé z autentizačních mechanismů EAP. Souběžně jsem taktéž nainstaloval několik programových balíčků, které rozšiřují základní funkčnost programu FreeRADIUS a mezi důležité patří:

1. *freeradius2-mysql*: Instaluje MYSQL modul pro FreeRADIUS server.
2. *freeradius2-ldap*: Instaluje LDAP modul pro FreeRADIUS server.
3. *freeradius2-utils*: Tento balíček zahrnuje simulační klientské nástroje jako *radclient*, *radtest*, *radzap*, *radsniff* a *smbencrypt*.
4. *freeradius-common*: Slouží k instalaci slovníků a příručkových souborů.
5. *libfreeradius2*: Obsahuje sdílené knihovny FreeRADIUS programu.
6. *libfreeradius-dev*: Obsahuje sdílené knihovny FreeRADIUS programu vyhrazené pro vývoj.

Další podstatný, již samostatný, program je *ssl-cert*, jenž umožňuje vytvoření certifikační autority a certifikátů potřebných pro některé typy autentizačních mechanismů EAP. Plný seznam instalovaných programů a knihoven je obsažen v skriptovacích souborech.

FreeRADIUS program je u operačních systémů Ubuntu spuštěný démonem *freerad*. Pro vypnutí běžící instance a následně spuštění programu FreeRADIUS v ladícím režimu použijte příkazy z výpisu 3. Řetězec *root@radiusserver:~#* není součástí jednotlivých příkazů, ale pouze značí informace o právech administrátora a názvu přihlášeného hostitele virtuálního stroje.

```
1 root@radiusserver:~# sudo -i
2 root@radiusserver:~# /etc/init.d/freeradius stop
3 root@radiusserver:~# freeradius -X
```

Výpis 3: Vypnutí procesu FreeRADIUS a spuštění ladícího režimu

Po spuštění programu FreeRADIUS v ladícím režimu se do příkazového řádku vypíše obsáhlý text s informacemi o inicializovaných modulech, spuštěných virtuálních instancích, definovaných klientech a zbylých údajích. Úspěšné spuštění programu FreeRADIUS je značeno textovým výstupem dle výpisu 4.

```
1 Listening on authentication address * port 1812
2 Listening on accounting address * port 1813
3 Listening on authentication address 127.0.0.1 port 18120 as server inner-tunnel
4 Listening on authentication address * port 2812 as server
      virtual_server_access_point
5 Listening on proxy address * port 1814
6 Ready to process requests.
```

Výpis 4: Textový výstup úspěšného spuštění programu FreeRADIUS

6.2 Struktura pracovního adresáře

Po instalaci programu FreeRADIUS lze nalézt hlavní pracovní adresář v */etc/freeradius/*. V případech, kdy se v textu vyskytuje název určitého souboru bez uvedené cesty, poté je za kořenový adresář považován */etc/freeradius/*. Funkčnost implementace vychází z obsahu konfiguračních souborů umístěných v tomto pracovním adresáři. Mezi důležité konfigurační soubory patří:

- *radiusd.conf*: Jedná se o hlavní konfigurační soubor, který obsahuje nastavení obecného typu v umístěných v různých sekcích. Dále obsahuje odkazy (formou *$INCLUDE*) na různé adresáře modulů a slovníků, které se mají inicializovat při spuštění.
- *clients.conf*: Pro definici různých RADIUS klientů (NAS) formou IP adres a sdílených hesel je použitý tento konfigurační soubor.
- */modules/ldap*: Adresář */modules/* obsahuje konfigurační soubory rozsáhlého počtu různých modulů. Název modulu vždy odpovídá vlastnímu konfiguračnímu souboru, jakož je to například u souboru *ldap*.
- *sql.conf*: I když je *sql* modul, tak jeho konfigurační soubor leží v kořenu pracovního adresáře.
- *eap.conf*: Tento konfigurační soubor se využívá pro nastavení určitých metod autentizačních mechanismů EAP.
- */sites-enabled/default*: Program FreeRADIUS dovolí využívat virtuálních instancí, které lze inicializovat s vlastní konfigurací. Pro vytvoření nové virtuální instance se definuje konfigurační soubor v adresáři */sites-available/* a napojí se symbiotickou vazbou do adresáře */sites-enabled/*. Mezi výchozí konfigurační soubory virtuálních instancí patří *default* pro obecné použití a soubor *inner-tunnel* pro EAP žádosti.
- *proxy.conf*: V tomto konfiguračním souboru se definují oblasti (angl. realms), mezi kterými jsou předávány RADIUS zprávy. RADIUS server se stává klientem pro jiný RADIUS server jiné oblasti v případě předávání zpráv.

- *users*: Konfigurační soubor *users* je výchozí textové úložiště pro definované uživatele.

6.3 Definice RADIUS klienta (NAS)

RADIUS klienti jsou na straně RADIUS serveru definováni v souboru *clients.conf*. RADIUS klientem může být například NAS, jiný RADIUS server jednající jako zástupce či přístupový bod. Ve výpisu 5 je znázorněna definice RADIUS klienta s názvem hostitele *radiusclient*. Za povinné parametry se považuje IP adresa virtuálního stroje, sdílené heslo mezi RADIUS serverem a klientem a typ NAS zařízení.

```
1 client radiusclient {
2    ipaddr = 192.168.1.112
3    secret= sharedpass123
4    nastype = other
5 }
```

Výpis 5: Definice RADIUS klienta v souboru *clients.conf*

Můžete si všimnout, že IP adresa je zapsána v číselném tvaru. Před začátkem návrhu implementace byly přiřazeny IP adresy k názvům hostitelů v souboru */etc/hosts*. Žádoucí je, aby se v konfiguračních síťových protokolech a službách odkazovalo na virtuální stroje v podobě hostitelského názvu či FQDN. Program FreeRADIUS bohužel obecně vyžaduje definici IP adresy v číselném tvaru, zatímco program k implementace Diameter protokolu (freeDiameter) již umožňuje využít název hostitele ve většině případů.

6.4 Definice uživatelských údajů

Pro definici uživatelských údajů je možné využít několik různých metod, kde každá má své výhody, nevýhody a jiná využití. Mezi v implementaci zahrnuté metody k definici uživatelských údajů a uložení účtovacích dat patří:

1. *users*: Jedná se o výchozí textový soubor pro definici uživatelských údajů. Příklad obsahu souboru *users* můžete vidět ve výpisu 6. Každý navazující řádek na řádek definující jméno uživatele musí být odsazen klávesou tabulátor. V daném výpisu můžete vidět tři definované uživatele a to *petr_files*, *marie_files* a *martina_files*. Výpis dále obsahuje definici AVP pro autorizační omezení časového rozmezí možného připojení mezi 05:00 až 23:00 pro uživatele *marie_files*. Pro uživatele *martina_files* je uživatelské heslo definováno v šifrovaném tvaru hašovací funkcí typu bezpečný hašovací algoritmus (SHA).

2. *MYSQL databáze*: Program FreeRADIUS se může připojit k SQL databázi pro dohledání uživatelských údajů při využití modulu *sql* a úpravy souboru *sql.conf*. Databáze typu MYSQL je oproti textovému souboru *users* škálovatelná, uživatelsky přívětivější s užitím webového rozhraní, editovatelná za provozu programu FreeRADIUS, více zabezpečená a

strukturovaná na základě uživatelských profilů či skupin. Mimo uložení uživatelských dat lze MYSQL databázi využít pro sběr účtovacích dat či alternativní definici RADIUS klientů namísto souboru *clients.conf*. Ve výpisu 7 je znázorněno vytvoření MYSQL databáze s názvem *radius_db* s datovou strukturou dle schématu ze souboru */sql/mysql/schema.sql* a přidání uživatele *petr_sql*.

3. *LDAP databáze*: LDAP je protokol pro přístup k úložišti přes TCP/IP síť. Pomocí LDAP databáze je možné autentizovat uživatele, spravovat digitální certifikáty či poskytovat informace o uzlech a zařízeních v síti [23]. V implementaci protokolu RADIUS jsem využil projektu OpenLDAP, jehož instalovatelný program se nazývá *slapd*. Datová struktura databáze je vytvořena pomocí schématu v souboru */etc/ldap/schema/freeradius.schema*. Pro naplnění LDAP databáze uživatelskými údaji lze využít skriptovací soubor */scripts/ldap_db_populate.ldif*, jehož částečný obsah můžete nalézt ve výpisu 8. Při nasazení LDAP databáze je možné číst uživatelské heslo přímo z databáze namísto vytváření vazby mezi uživatelskými údaji a LDAP serverem. Toto řešení poté přináší urychlení procesu přečtení a ověření uživatelských údajů.

```
1 "petr_files" Cleartext-Password := "petr_files_heslo"
2   Reply-Message = "Uzivatel %{User-Name} zadal o pristup."
3
4 "marie_files" Cleartext-Password := "marie_files_heslo", Login-Time := Al0500
    -2300
5   Reply-Message = "Uzivatel %{User-Name} zadal u pristup z NAS adresy %{NAS-IP
      -Address}."
6
7 "martina_files" SHA-Password := "OcMD027C0aBX7KnSoMyPhOoR5Mo="
8   Reply-Message = "Uzivatel %{User-Name} zadal o pristup."
```

Výpis 6: Vzorová definice uživatelů v souboru *users*

```
1 root@radiusserver:~# mysqladmin -u root -p create radius_db
2 root@radiusserver:~# mysql -u root -p radius_db < /etc/freeradius/sql/mysql/
    schema.sql
3 root@radiusserver:~# mysql -u root -p radius_db
4 INSERT INTO radcheck (username, attribute, op, value) VALUES ('petr_sql', '
    Cleartext-Password', ':=', 'petr_sql_heslo');
5 INSERT INTO radreply (username, attribute, op, value) VALUES('petr_sql', 'Reply
    -Message', '=', 'Uzivatel petr_sql zadal o pristup.');
```

Výpis 7: Vzorové vložení uživatelských údajů do MYSQL databáze

```
dn: cn=marie_ldap,ou=uzivatele,ou=radius,dc=vsb,dc=cz
objectclass: radiusProfile
objectClass: person
cn: marie_ldap
sn: marie_ldap
userPassword: marie_ldap_heslo
description: Uzivatel marie_ldap s heslem marie_ldap_heslo v nesifrovanem tvaru
radiusGroupName: studenti
radiusExpiration: "1 January 2005"
radiusReplyMessage: "Uzivatel marie_ldap, ulozeny v LDAP databazi, zadal o
    pristup."
```

Výpis 8: Vzorová část obsahu souboru *ldap_db_populate.ldif*

6.5 Šifrování uživatelských hesel

Uživatelské hesla lze ukládat v šifrované podobě. Pro generování šifrovaných hesel jsem vytvořil skriptovací soubory spustitelné v interpretru *Perl*, které jsou umístěné v adresáři */scripts/*. Různé z metod šifrování uživatelských hesel již byly popsány v kapitole 2.1. Ve výpisu 9 můžete vidět generování šifrovaného hesla užitím hašovací funkce typu SHA pro uživatele *martina_files*. Mezi další metody šifrování hesel patří i hašovací funkce typu MD5 a obě varianty s přidanou kryptografickou solí. Šifrování hesel přináší určitou výhodu tehdy, kdy existuje bezpečnostní riziko ukládat uživatelské hesla v nešifrovaném tvaru.

V programu FreeRADIUS existuje určité omezení při ukládání hesel v šifrovaném tvaru. V kapitole 3.1 bylo řečeno, že RADIUS umožňuje podporu několik autentizačních protokolů jako PAP, EAP, CHAP či MS-CHAP. V případě využití jiné metody než PAP je nutné ukládat uživatelská hesla v nešifrovaném tvaru (AVP typu *Cleartext-Password*) pro zajištění správné funkčnosti.

```
root@radiusserver:~# perl /etc/freeradius/scripts/password_sha1_generate.pl
    martina_files_heslo

OcMD027C0aBX7KnSoMyPhOoR5Mo=
```

Výpis 9: Způsob šifrování uživatelských hesel

6.6 Simulační nástroje pro ověření funkčnosti

Při návrhu implementace protokolů RADIUS a Diameter je vhodné vycházet z obecné filozofie, kterou lze shrnout v těchto bodech:

1. Provádět minimální změny v konfiguračních souborech.

2. Ověřit funkčnost každé provedené změny v řadě případů.

3. Ukládat si funkční stavy například způsobem záchytných bodů.

Programový balíček *freeradius2-utils*, který je součástí této implementace protokolu RADIUS, obsahuje několik simulačních nástrojů vhodných pro ověření funkčnosti konfigurace. Mezi instalované nástroje patří *radtest*, *radclient* a další. Ve výpisu 10 je znázorněný způsob ověření stávající konfigurace vůči lokálnímu hostiteli (angl. *localhost*) a konkrétně autentizace uživatele *petr_files* s uživatelským heslem *petr_files_heslo*, žádajícího o vstup z NAS portu *100* při užití sdíleného hesla *sharedpass123*. Žádost o přístup byla schválena RADIUS serverem.

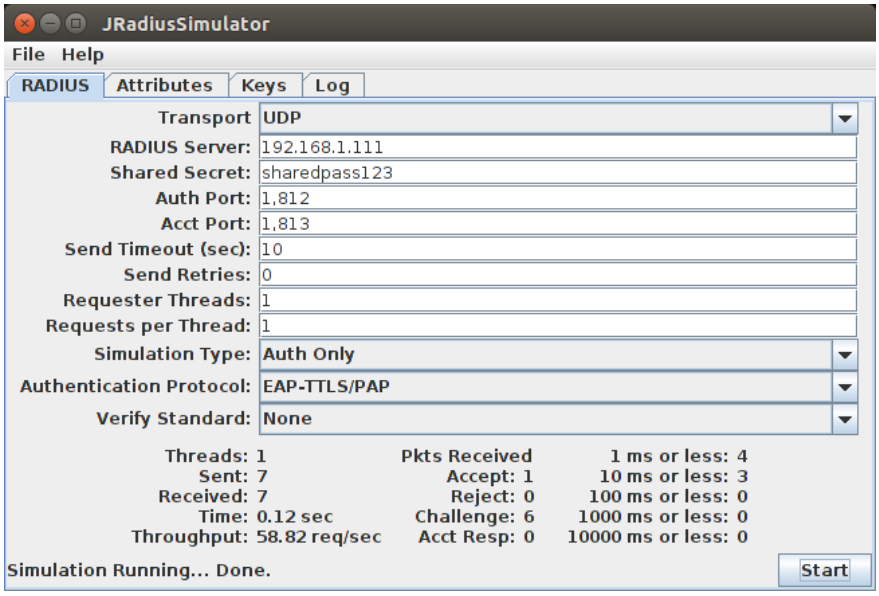
```
root@radiusserver:~# radtest petr_files petr_files_heslo localhost 100
    sharedpass123

Sending Access-Request of id 199 to 127.0.0.1 port 1812
User-Name = "petr_files"
User-Password = "petr_files_heslo"
NAS-IP-Address = 192.168.1.111
NAS-Port = 100
Message-Authenticator = 0x00000000000000000000000000000000
rad_recv: Access-Accept packet from host 127.0.0.1 port 1812, id=199, length=58
Reply-Message = "Uzivatel petr_files zadal o pristup."
```

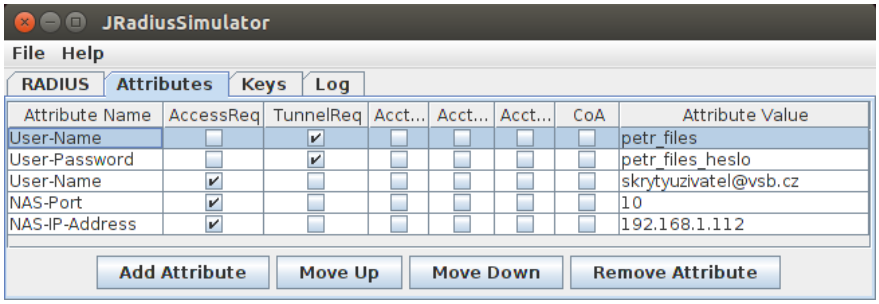
Výpis 10: Využití simulačního nástroje *radtest*

V kapitole 5.4 bylo řečeno, že každý skriptovací soubor obsahuje sadu číslovaných příkazů pro lokální a vzdálené ověření stávající konfigurace. Pro zjištění úplné syntaxe simulačních příkazů můžete otevřít text manuálu například příkazem *man radtest*.

Při instalaci RADIUS klienta (NAS) existuje také možnost zprovoznit simulační nástroj *JRadius Simulator* s grafickým rozhraním. Nástroj *JRadius Simulator* rozšiřuje nástroj *radtest* o funkci vložení klientských certifikátů a certifikační autority. Příklad vyplněných údajů pro ověření funkčnosti simulace můžete vidět na obrázcích 6.6.1 a 6.6.2. V příkladu ověření funkčnosti konfigurace byl použitý autentizační protokol typu EAP-TTLS, kdy RADIUS zpráva Žádost-Přístupu obsahuje vnitřní a vnější identitu uživatele. Vnější identita je viditelná okolnímu světu a lze ji zjistit odposlechem. V případě konfigurace určitých domén je nutné do fráze vnější identity zahrnout i skutečné jméno domény. Skrytá identita není přenášena v otevřeném tvaru a je tedy chráněna vůči odposlechům.



Obrázek 6.6.1: Nastavení síťových parametrů v nástroji *JRadius Simulator*



Obrázek 6.6.2: Nastavení obsahu AVP v nástroji *JRadius Simulator*

6.7 Nastavení obecného chování programu FreeRADIUS

Pro konfiguraci obecného chování programu FreeRADIUS se využívá úprava souboru *default*, jenž je umístěn symbiotickou vazbou v adresářích */sites-available/* i */sites-enabled/*. V kapitole 6.2 bylo řečeno, že konfigurační soubor *default* je výchozí virtuální instance pro obecné využití. Každý z virtuálních serverů může obsahovat několik sekcí, mezi které patří *client*, *authorize*, *authenticate*, *preacct*, *accounting*, *session*, *post-auth*, *pre-proxy* a *post-proxy*. Určení pořadí jednotlivých sekcí je podstatné pro správnou funkčnost implementace. Pro každou ze sekcí se typicky definuje jeden či více modulů (např. *chap*, *files*, *sql* či *ldap*) které převezmou a zpracují obsah obdržené RADIUS zprávy Žádost-Přístupu či Žádost-Účtování. Obsah RADIUS zprávy se sekvenčně zpracovává v těchto sekcích:

1. *authorize*: Jedná se o sekci, jenž program FreeRADIUS spouští v prvním pořadí. Vhodnější název sekce by vystihovala fráze identifikace namísto autorizace. Účelem této sekce je identifikovat typ zprávy Žádost-Přístupu a dále identifikovat profil uživatele. Na základě

výstupu této sekce se rozhoduje o způsobu, jak zprávu Žádost-Přístupu dále předat k vyhodnocení.

2. *authenticate*: Tato sekce správně vystihuje proces autentizace, kdy dochází k ověření uživatelských dokladů vůči identitě uživatele.

3. *accounting*: V této sekci se určí modul, který má vyhodnotit zprávu Žádost-Účtování a zprostředkovat uložení účtovacích dat.

4. *post-auth*: Poslední jmenovanou sekci vystihuje správně pojem autorizace. Jelikož byl uživatelský profil identifikován sekcí *authorize* a rozhodnutí o autentizace zajistila sekce *authenticate*, je tedy možné uživatele autorizovat na základě určité politiky.

Pro formulaci autorizační politiky je vhodné využít speciálního jazyku *unlang*, ve kterém lze jednoduše vyjádřit chování programu FreeRADIUS formou podmíněných příkazů. Ve výpisu 11 je vyjádřeno, jakým způsobem lze definovat autorizační politiku v sekci *post-auth* konfiguračního souboru *default*.

První podmíněný příkaz jazyku *unlang* ve výpisu 11 je typově informativní namísto autorizační. Na základě hodnoty AVP typu *NAS-Port* v obsahu RADIUS zprávy Žádost-Přístupu je upravena hodnota AVP *Odpověď-Zprávy* (angl Reply-Message), která obsahuje formátovaný výstup s informací o využitém portu NAS, dotazujícího se uživatele a aktuálního času žádosti. Druhý podmíněný příkaz provede zamítnutí zprávy Žádost-Přístupu v případě, když je uživatel definovaný MYSQL databází a dále je zařazen ve skupině *studenti*. Na závěr je taktéž vypsán formátovaný výstup s informací o zamítnutí přístupu.

```
1  post-auth {
2     if(request:NAS-Port > 100){
3        update reply {
4           Reply-Message := "Byl vyuzit NAS port c. %{NAS-Port} udeleni pristupu
                uzivateli %{User-Name} v case %{sql:SELECT curtime();}."
5        }
6     }
7     if(SQL-Group == studenti){
8        update reply {
9           Reply-Message := "Studentum neni udelen pristup!"
10       }
11       reject
12    }
13 }
```

Výpis 11: Formulace autorizační polity jazykem *unlang* v souboru *default*

6.8 Konfigurace přístupového bodu

Velmi užitečným nástrojem pro rozšíření stávající odladěné konfigurace programu FreeRADIUS je využití virtuálních instancí. V kapitole 6.2 již byly virtuální instance obecně popsány. Pro demonstraci funkce virtuálních instancí jsem do síťového schématu zavedl i přístupový bod (*radiusaccesspoint*), jenž plní funkci druhého RADIUS klienta a musí být nejprve definovaný v souboru *clients.conf*. Definici přístupového bodu můžete vidět ve výpisu 12.

```
client radiusaccesspoint {
   secret = accesspoint123
   virtual_server = virtual_server_access_point
}
```

Výpis 12: Definice přístupového bodu v souboru *clients.conf*

Pro virtuální instanci přístupového bodu je možné definovat vlastní modul, jenž určuje textový soubor obsahující uživatelské údaje. Definice modulu */modules/files_access_point* je znázorněna ve výpisu 13, na jehož základě je možné specifikovat uživatele *jindrich_virtual* v souboru *users_acesss_point*, který je vyhrazený pouze pro RADIUS přístupový bod.

Na závěr se musí vytvořit soubor *virtual_server_access_point* dané virtuální instance přístupového bodu v adresáři */sites-available/* se symbiotickou vazbou do adresáře */sites-enabled/*. Soubor *virtual_server_access_point* v sobě obsahuje definici naslouchaných IP adres a portů v sekci *listen*, informace o síťových parametrech virtuálního stroje *radiusaccessserver* a sekci *authorize*, jejíž částečný obsah můžete vidět ve výpisu 14. Definice autorizační politiky, která již byla popsána v kapitole 6.7, není vyhrazena pouze pro sekci *post-auth*.

```
files files_access_point{
   usersfile = /etc/freeradius/users_access_point
   compat = no
}
```

Výpis 13: Definice vlastních uživatelů v souboru */modules/files_access_point*

Záměrem konfigurace přístupového bodu bylo demonstrovat způsob rozšíření stávající implementace využitím virtuálních instancí. V případě, kdy není identifikován uživatelský profil sekcí *authorize* (uživatelské jméno se liší od *jindrich_virtual*), dojde k zpracování podmíněného příkazu jazyka *unlang*. V časovém rozmezí mezi 9:00 - 11:00 libovolného dne jsou povoleny všechny žádosti o přístup od neznámých uživatelů, kteří se připojili přes virtuální stroj přístupového bodu (*radiusaccesspoint*).

```
files_access_point
if(noop){
   update control {
      Login-Time := Al0900-1100
```

```
5       Auth-Type := "Accept"
6    }
7    update reply {
8       Reply-Message := "Neznamemu uzivateli %{User-Name} byl udelen pristup ve
            vyhrazenem case 09:00 - 11:00."
9    }
10 }
```

Výpis 14: Definice přístupového bodu v souboru *clients.conf*

6.9 Využití domén a RADIUS zástupců

Domény se využívají za účelem třídění uživatelů typicky na základě názvu serveru ze systému doménových jmen (DNS), který je uživateli přiřazený. Název serveru DNS je od uživatelského jména oddělen speciálním znakem, kterým je většinou znak „@“. Plné uživatelské jméno s přiřazenou doménou může například být *dusan_files@vsb.cz*. Pro běžné použití v rámci jednotné domény není nutné specifikovat doménovou část při žádosti o přístup.

Do síťového schématu konfigurace protokolu RADIUS jsem zařadil celkem dva RADIUS servery, které můžete vidět na obrázku 5.2.1. RADIUS server *radiusserver* je hlavním serverem, jehož doménou je Vysoká škola Báňská (VŠB). Druhý RADIUS server *radiusproxy* má vyhrazenou doménu pro Vysoké učení technické v Brně (VUT). V kapitole 3.9 bylo řečeno, že se RADIUS zástupce (angl. proxy) chová jako RADIUS server v případě řešení AAA procesů uživatele své domény, zatímco jako RADIUS klient v případě, kdy uživatel spadá do domény jiné a RADIUS zprávy musí být předány.

Dle síťového schématu jsou všechny žádosti uživatelů VŠB či žádné domény zpracovány RADIUS serverem *radiusserver*, kdežto žádosti od uživatelů z domény VUT budou předány k vyhodnocení vůči RADIUS serveru *radiusproxy*. Každý z RADIUS serverů musí mít definované síťové parametry svých RADIUS zástupů v souboru *clients.conf* obdobným způsobem, jako byl definovaný klient přístupového bodu ve výpisu 12. Dále se v souboru *proxy.conf* definují informace o RADIUS zástupci (konkrétně *radiusproxy*), jejíž částečný obsah je můžete vidět ve výpisu 15. Pro doménu VUT je definovaný pouze jeden RADIUS zástupce s konkrétní IP adresou a sdíleným heslem, který má vyhrazeno převzít autentizační a účtovací žádosti. Konfigurační soubor *proxy.conf* musí vždy obsahovat i definici domény lokálního hostitele vlastního virtuální stroje, která je součástí skriptovacích souborů v příloze práce.

```
1  home_server home_server_vut.cz {
2     type = auth+acct
3     ipaddr = 192.168.1.113
4     port = 1812
5     secret = proxypass123
```

```
6  }
7  home_server_pool pool_vut.cz {
8     type = fail-over
9     home_server = home_server_vut.cz
10 }
11 realm vut.cz {
12    pool = pool_vut.cz
13    nostrip
14 }
```

Výpis 15: Část definice RADIUS zástupce *radiusproxy* v souboru *proxy.conf*

6.10 Zprovoznění RADIUS klienta (NAS)

V předchozích kapitolách praktické části byla konfigurována pouze část RADIUS serveru pro doménu VŠB, tedy virtuální stroj *radiusserver*, zatímco součástí této kapitoly je provést konfiguraci RADIUS klienta *radiusclient*.

Pro funkci klientské části RADIUS protokolu není nutné instalovat program FreeRADIUS, který je nasazený na serverové části. Za účelem ověření funkčnosti je pouze nainstalovaný simulační nástroj *freeradius-utils* a nepovinný program *JRadius Simulator*. Dále jsem do adresáře */scripts/* umístil skriptovací soubory pro ověření účtování ve stejném tvaru jako pro RADIUS server. Velmi užitečným způsobem konfigurace RADIUS klienta je instalace PAM, neboli zásuvných autentizačních modulů. Při instalaci konkrétního PAM je RADIUS klientovi umožněno generovat RADIUS zprávy v případě obdržení žádosti o přístup od některého z uživatelů.

Při konfiguraci RADIUS klienta je nainstalováno několik služeb, ke kterým může uživatel žádat o přístup. Pro obecná zařízení NAS platí, že zprostředkují udělení přístupu na základě výměny RADIUS zpráv mezi RADIUS serverem a poté přesměrují uživatele k zařízení s žádanou službou. Pro zjednodušení síťového schématu jsem umístil instalaci služeb přímo na zařízení NAS, respektive RADIUS klient *radiusclient*. Uživatel, který se připojuje do sítě za účelem využití některé z instalovaných služeb, musí nejprve kontaktovat RADIUS klienta, který započne výměnu RADIUS zpráv s RADIUS serverem. Na základě vyhodnocení autentizace a autorizace RADIUS serverem bude uživatelova žádost buď schválena či zamítnuta. Mezi instalované služby, kterých může uživatel využít, patří SSH, SUDO (příkaz pro přihlášení jako administrátor) a úložiště dat užitím FTP. Pro instalaci PAM s podporou protokolu RADIUS je potřeba buď nainstalovat balíček *libpam-radius-auth* užitím programového repositáře nebo rozbalit a sestavit balíček *pam_radius-1.3.17-ipv6.tar.gz* z externího zdroje, který již podporuje IPv6. V této kapitole budu vycházet z konfigurace pro verzi IPv4 a řešení kompatibilní s IPv6 můžete nalézt v skriptovacím souboru *freeradius_install_client.sh*. Konfigurace pro IPv4 a IPv6 se odlišují pouze ve způsobu instalace, umístění konfiguračních souborů a definicí IP adres. Po instalaci PAM se vytvoří konfigurační soubor */etc/pam_radius_auth.conf*, do kterého se musí definovat

síťové parametry RADIUS serveru dle výpisu 16. Při spuštění PAM modulu je tedy RADIUS klient odkázán na RADIUS server, který rozhoduje o autentizaci a autorizaci uživatele.

```
1 # server[:port]    shared_secret      timeout (s)
2 127.0.0.1          secret             3
3 radiusserver       sharedpass123      5
```

Výpis 16: Konfigurace PAM vůči RADIUS serveru v souboru */etc/pam_radius_auth.conf*

Výchozí PAM adresář pro nastavení jednotlivých služeb je umístěný v */etc/pam.d/*. Například konfigurace PAM pro službu FTP je vyznačena ve výpisu 17. Výpis obsahuje úplnou konfiguraci souboru */etc/pam.d/vsftpd* a záměrně jsem v textu ponechal neaktivní řádky kódu začínající znakem „#“. Pro využití služby FTP je obecně nutné, aby byl uživatel definovaný v systému daného virtuálního stroje *common-account*. V aktuální konfiguraci je definovaný jediný dostačující autentizační faktor ověření PAM *pam_radius_auth.so*, jehož konfigurační soubor se odkazuje RADIUS server.

```
1 auth    sufficient    pam_radius_auth.so
2 #auth   required      pam_listfile.so item=user sense=deny file=/etc/ftpusers
3 #auth   required      pam_shells.so
4 #@include common-account
5 #@include common-session
6 #@include common-auth
```

Výpis 17: Definice PAM pro službu FTP v souboru */etc/pam.d/vsftpd*

Pro ověření funkčnosti se můžete připojit z virtuálního stroje AAA uživatel (*aaauser*) zadáním příkazů do příkazového řádku dle výpisu 18.

```
1 root@aaauser:~# ftp radiusclient
2 root@aaauser:~# petr_files@vsb.cz
3 root@aaauser:~# petr_files_heslo
```

Výpis 18: Příklad využití FTP z virtuálního stroje *aaauser*

Uživatel *petr_files* z domény VŠB si vyžádal přístup ke službě při kontaktování RADIUS klienta. RADIUS klient dále navázal spojení s RADIUS serverem, který uživatele autentizoval vůči souboru *users*, autorizoval dle vlastní politiky a odeslal schválení žádosti zpět RADIUS klientovi. Uživateli *petr_files* se po schválení žádosti zobrazí jeho osobní adresář, kde si může stáhnout textový soubor obsahující pozdrav užitím příkazu *get pozdrav.txt*.

Na závěr je na RADIUS klienta nainstalován program Hostapd, kterým lze zprovoznit přístupový bod do typicky bezdrátové, ale i pevné sítě. Hostapd je serverová část pro klienta s názvem wpa_supplicant, jenž je nainstalován na virtuálním stroji *aaauser* a bude popsán v navazující kapitole. Pro výměnu informací mezi programy Hostapd a wpa_supplicant se v bezdrátových

sítích (Wi-Fi) využívá chráněný přístup k Wi-Fi (WPA). Pro výměnu informací v pevných sítích je možno využít některé z typu metody EAP.

Konfiguračním souborem programu Hostapd je */etc/hostapd/hostapd.conf*, jehož úplný obsah můžete vidět ve výpisu 19. Konfigurace obsahuje definici pevného připojení přes ethernetové rozhraní názvu *eth0*, identifikátor přístupového bodu *radius.ap.vsb.cz* a síťové parametry RADIUS serveru. Uživatel se do pevné sítě připojí s využitím klientského programu wpa_supplicant a jeho uživatelské doklady budou ověřeny vůči RADIUS serveru. Při úspěšné autentizaci a autorizace bude uživatel připojen do sítě sjednáním komunikace mezi RADIUS klientem *radiusclient* a AAA uživatelem *aaauser*.

```
1  interface=eth0
2  driver=wired
3  use_pae_group_addr=1
4  ieee8021x=1
5  eap_reauth_period=3600
6  nas_identifier=radius.ap.vsb.cz
7  #radius_server_ipv6=1
8  own_ip_addr=192.168.1.112
9  auth_server_addr=192.168.1.111
10 auth_server_port=1812
11 auth_server_shared_secret=sharedpass123
12 acct_server_addr=192.168.1.111
13 acct_server_port=1813
14 acct_server_shared_secret=sharedpass123
```

Výpis 19: Konfigurace programu Hostapd v souboru */etc/hostapd/hostapd.conf*

Příkaz pro spuštění programu Hostapd na virtuálním serveru *radiusclient* a textový výstup do příkazového řádku úspěšného zprovoznění přístupového bodu *ap.radius.vsb.cz* obsahuje výpis 20.

```
1  root@radiusclient:~# hostapd /etc/hostapd/hostapd.conf
2
3  Configuration file: /etc/hostapd/hostapd.conf
4  Using interface eth0 with hwaddr 00:0c:29:e5:98:86 and ssid ""
5  eth0: RADIUS Authentication server 192.168.1.111:1812
6  eth0: RADIUS Accounting server 192.168.1.111:1813
7  eth0: interface state UNINITIALIZED->ENABLED
8  eth0: AP-ENABLED
```

Výpis 20: Úspěšné spuštění programu Hostapd z virtuálního stroje *radiusclient*

6.11 Zprovoznění AAA uživatele

Pro využití služeb SSH, SUDO a FTP uživatel z virtuální stroje *aaauser* není nutné instalovat a konfigurovat dodatečné programy. Pro zprostředkování bezpečné komunikace s programem Hostapd, který byl konfigurován v předchozí kapitole, je potřebné nastavit suplikant s názvem wpa_supplicant.

Konfigurační soubor programu wpa_supplicant je */etc/wpa_supplicant/radius_vsb_wpa.conf*, jehož úplný obsah zahrnuje výpis 21. Výpis zahrnuje definici uživatelských údajů a určení metody EAP typu PEAP.

```
network={
   ssid="radius.ap.vsb.cz"
   key_mgmt=WPA-EAP
   pairwise=TKIP
   group=TKIP
   eap=PEAP
   identity="petr_files"
   password="petr_files_heslo"
   phase1="peapver=0"
   phase2="MSCHAPV2"
}
```

Výpis 21: Konfigurace programu wpa_supplicant na virtuálním stroji *aaauser*

Pro vytvoření zabezpečené komunikace s RADIUS klientem je potřebné zadat příkaz, jehož obsah společně s textovým výstupem úspěšného spojení můžete najít ve výpisu 22.

```
root@aaauser:~# wpa\_supplicant -i eth0 -c /etc/wpa\_supplicant/radius_vsb_wpa.
    conf -D wired

Successfully initialized wpa\_supplicant
eth0: Associated with 01:80:c2:00:00:03
eth0: CTRL-EVENT-EAP-STARTED EAP authentication started
eth0: CTRL-EVENT-EAP-METHOD EAP vendor 0 method 25 (PEAP) selected
EAP-MSCHAPV2: Authentication succeeded
EAP-TLV: TLV Result - Success - EAP-TLV/Phase2 Completed
eth0: CTRL-EVENT-EAP-SUCCESS EAP authentication completed successfully
```

Výpis 22: Sestavení komunikace mezi užitím programu Hostapd a wpa_supplicant

Velmi prospěšné je sledovat textový výstup hlavního virtuálního serveru *radiusserver* při spuštěném programu FreeRADIUS v ladícím režimu, který byl popsán v kapitole 6.1. Informace o schválení žádosti o přístup uživatele *petr_files* obsahuje výpis 23. V textu práce není zahrnutý

celkový textový výstup z ladícího režimu programu FreeRADIUS, který dále zahrnuje periodickou výměnu účtovacích zpráv Žádost-Účtování a Odpověď-Účtování, vyjednání tunelované komunikace a seznam modulů a virtuálních instancí programu FreeRADIUS, které zprostředkovaly ověření uživatele *petr_files*.

```
1 Sending Access-Accept of id 11 to 192.168.1.112 port 33742
2    Reply-Message = "Uzivatel petr_files zadal o pristup."
3    User-Name = "petr_files"
4    MS-MPPE-Recv-Key = 0
         x4c347acd48686fd7630a730b519c4c3e698313e090d0dc3b6f01c09b39d85974
5    MS-MPPE-Send-Key = 0
         xd16f88c58b196be7f213b4a8b279ba054694393781944ef8e1936a54329d6af3
6    EAP-Message = 0x03de0004
7    Message-Authenticator = 0x00000000000000000000000000000000
8 Finished request 12.
```

Výpis 23: Schválení žádosti o přístup ve výstupu ladícího režimu programu FreeRADIUS

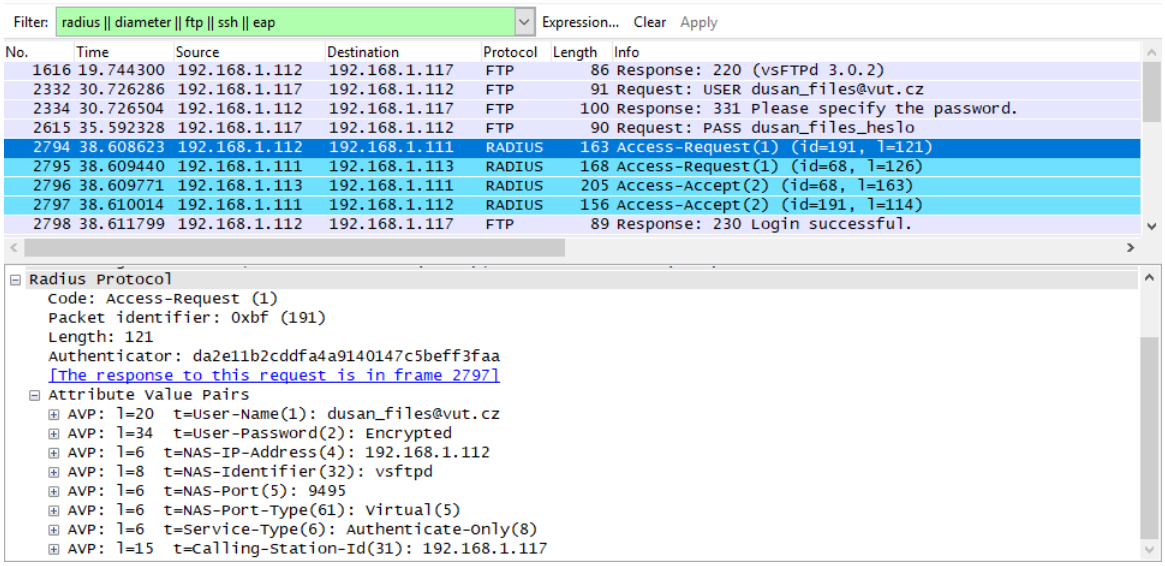
6.12 Analýza síťové komunikace v prostředí Wireshark

V kapitole 5.4 bylo řečeno, že skriptovací soubory taktéž obsahují sadu číslovaných příkazů pro ověření funkčnosti implementace protokolů RADIUS a Diameter. Několika číslovaným příkazům odpovídá i záznam síťové komunikace na ethernetovém rozhraní *eth0*. Záznam komunikace je soubor s příponou *.pcap*, jenž lze otevřít v protokolovém a paketovém analyzátoru Wireshark, který lze volně získat ze stránek výrobce programu [24].

Například pro analýzu schválení přístupu uživatele *dusan_files@vut.cz*, který žádá o přístup k FTP úložišti z virtuálního stroje *aaauser*, můžete otevřít záznam komunikace s názvem *aaauser_5.pcap*. Na obrázku 6.12.1 je vyobrazeno prostředí Wireshark při otevření záznamu komunikace ze souboru *aaauser_5.pcap*. Na ethernetovém rozhraní *eth0* se přenáší velké množství síťových dat různých protokolů a je tedy vhodné využít protokolových filtrů. Pro analýzu v prostředí Wireshark jsem zvolil filtrování dle protokolů RADIUS, Diameter, FTP, SSH a EAP („radius || diameter || ftp || ssh || eap“).

Ze záznamu komunikace na obrázku 6.12.1 lze zjistit, že virtuální stroj *aaauser* s IP adresou 192.168.1.117 předal uživatelské údaje virtuálnímu stroji s IP adresou 192.168.1.112 (*radiusclient*) za účelem využití služby FTP. Zde můžete vidět záměrný nedostatek v zabezpečení uživatelských údajů, které jsou předány v nešifrovaném tvaru a náchylné vůči odposlechu. Základní verze FTP totiž neumožnující zajistit jakoukoli formu šifrování komunikace a je proto vhodnější využít například FTP s podporou SSL/TLS (FTPS) či jiný protokol pro přenos souborů. Po předání uživatelských údajů se naváže výměna RADIUS zpráv mezi virtuálním strojem *radiusclient* a virtuálním strojem *radiusserver* s IP adresou 192.168.1.111. Jelikož uživatel *dusan_files@vut.cz*

spadá do jiné domény, RADIUS server VŠB domény jedná jako zastupující klient a zasílá zprávu Žádost-Přístupu vůči virtuálnímu stroji *radiusproxy* s IP adresou 192.168.1.113. RADIUS server VUT domény vyhodnotí žádost o přístup dle vlastní uživatelské databáze i autorizační politiky a odešle zprávu Udělení-Přístupu vůči RADIUS serveru VŠB domény, který zprávu dále předá RADIUS klientovi. Žádost i přístup ke službě FTP byla schválena a uživatel *dusan_files@vut.cz* se připojí do svého osobního adresáře.



Obrázek 6.12.1: Analýza záznamu komunikace *aaauser_5.pcap* v prostředí Wireshark

Implementace protokolu RADIUS obsahuje řadu definovaných uživatelů, kteří se navzájem odlišují typem uživatelské databáze, formy šifrování uživatelských hesel, formátovaným výstupem v AVP typu *Odpověď-Zprávy* a autorizační politikou. Některým uživatelům je schválený či odepřený přístup na základě platnosti uživatelského hesla, přiřazené uživatelské skupiny, zvoleného portu NAS nebo vymezeného časového rozmezí.

Pro rozšiřující zavedení do problematiky implementace protokolů RAIDUS a Diameter opět doporučuji přečíst obsah skriptovacích souborů a otevřít zachycenou komunikaci v prostředí Wireshark.

7 Implementace protokolu Diameter

7.1 Instalační program freeDiameter

K návrhu implementace protokolu Diameter jsem zvolil program freeDiameter 1.2.1, který je možné volně získat v podobě zdrojového kódu z internetových stránek vývojáře programu [25]. Projekt freeDiameter je implementace protokolu Diameter vycházející z doporučení RFC 3588 [15] (nyní nahrazeno doporučením RFC 6733 [16]), které definuje základní specifikaci protokolu Diameter.

Program freeDiameter se skládá z těchto komponent:

- *libfdproto*: Je společná knihovna zajišťující funkce k manipulaci Diameter zpráv a slovníků. Záměrem knihovny je její opětovné použití v jiných projektech, které by vyžadovaly manipulaci či překlad Diameter zpráv.
- *libfdcore*: Tato sdílená knihovna obsahuje jádro rámce programu freeDiameter. Knihovna zřizuje síťové připojení s jinými Diameter účastníky a provádí úlohy definované v základní specifikaci protokolu Diameter, mezi které například patří hlídací mechanismy, základní směrování a spouštění zásuvných modulů.
- *freeDiameterd*: Jedná se o jednoduchý démon, který zpracovává příkazové vstupy a inicializuje rámec programu freeDiameter. Démon se využívá ke spuštění programu freeDiameter na Diameter serveru, Diameter klientech a různých agentech.
- *extensions*: Tento komponent poskytuje prostředky k rozšíření funkčnosti rámce programu freeDiameter. Zásuvné moduly zajišťují manipulaci s aplikací Diameter serveru, pokročilé směrovací funkce a správu účastníků. [26]

Instalaci programu freeDiameter není možné provést z repositáře balíčků, jako tomu bylo u programu FreeRADIUS. Pro instalaci programu freeDiameter je možné využít nástroje *Mercurial* a postup instalace je znázorněn ve výpisu 24.

```
1 root@diamserver:~# hg clone http://www.freediameter.net/hg/freeDiameter
2 root@diamserver:~# mkdir fDbuild
3 root@diamserver:~# cd fDbuild
4 root@diamserver:~# cmake ../freeDiameter
5 root@diamserver:~# sed -i -e 's%ALL_EXTENSIONS:BOOL=OFF%ALL_EXTENSIONS:BOOL=ON%
      g' CMakeCache.txt
6 root@diamserver:~# make
7 root@diamserver:~# make install
```

Výpis 24: Instalace programu freeDiameter užitím nástroje *Mercurial*

Příkaz *hg clone* vytvoří kopii zdrojového adresáře programu freeDiameter z internetových stránek vývojáře [25]. Po kompilaci zdrojového adresáře příkazem *cmake* je potřebné upravit soubor *CmakeCache.txt*, který definuje pokyny pro instalaci. Instalace programu freeDiameter se všemi zásuvnými moduly se provede úpravou tvaru parametru *ALL_EXTENSIONS:BOOL=ON*. Dokončení samotné instalace se provede příkazy *make* a *make install*. Program freeDiameter se nainstaluje do výchozího pracovního adresáře */usr/local/etc/freeDiameter/*, který budu v navazujícím textu opět považovat za kořenový.

Pro některé z operačních systémů je nutné nastavit cestu k nově nainstalovaným sdíleným knihovnám, které jsou vyžadovány pro funkci programu freeDiameter dle výpisu 25

```
1 root@diamserver:~# LD_LIBRARY_PATH=/usr/local/lib
2 root@diamserver:~# ldconfig
```

Výpis 25: Nastavení cesty ke sdíleným knihovnám

7.2 Vytvoření SSL certifikátů

V kapitole 4.9.5 bylo řečeno, že podpora TLS mezi účastníky je nepovinná, ale před úspěšným spuštěním programu freeDiameter musí každý účastník předložit vlastní SSL certifikát, který se ověří vůči důvěryhodné certifikační autoritě. Tento způsob ověření již nabízí vyšší způsob zabezpečení, než je požadováno u programu FreeRADIUS, kde jsou účastníci (respektive server a klient) ověřeni pouze sdíleným heslem.

Pro vytvoření takzvaně sebou podepsaných certifikátů (angl. self-signed), které umožní spuštění programu freeDiameter bez nutnosti komunikace užitím TLS, je možné využít příkazů z výpisu 26. Použitím programu *openssl* se do adresáře */diametercert/* vytvoří celkem tři soubory. Privátní certifikát *privkey.pem* slouží pro ověření identity lokálního účastníka, soubor *cert.pem* obsahuje seznam důvěryhodných certifikačních autorit a soubor *dh.pem* definuje výměnu klíčů. Parametr běžného jména (CN) privátního certifikátu musí odpovídat hodnotě FQDN daného virtuální stroje, kterému certifikát náleží. Platnost certifikátu je stanovena na deset let.

```
1 root@diamserver:~# mkdir -p /usr/local/etc/freeDiameter/diametercert/
2 root@diamserver:~# cd /usr/local/etc/freeDiameter/diametercert/
3 root@diamserver:~# openssl req -new -batch -x509 -days 3650 -nodes \-newkey rsa
    :1024 -out cert.pem -keyout privkey.pem \-subj /CN=diamserver.vsb.cz
4 root@diamserver:~# openssl dhparam -out dh.pem 1024
```

Výpis 26: Vytvoření SSL certifikátů pro virtuální stroj *diamserver*

7.3 Využití vzorových konfiguračních souborů

Protokol Diameter pracuje na bázi rovnocenných účastníků, jak již bylo řečeno v kapitole 4.5. Pro zprovoznění Diameter serveru, klienta či různých agentů lze tedy použít totožný program freeDiameter, který se liší pouze v konfiguraci.

Instalační adresář obsahuje řadu vzorových konfiguračních souborů a jejich přesunutí do pracovního adresáře znázorňuje výpis 27. Hlavním konfiguračním souborem programu freeDiameter je *freeDiameter.conf* a konfigurační soubory zásuvných modulů obsahuje adresář */extensions/samples/*. Vzorové konfigurační soubory byly takto přesunuty na obou virtuálních strojích *diamserver* a *diamclient* ze síťového schématu dle obrázku 5.2.2.

```
1 root@diamserver:~# cp /install_files/freediameter/freeDiameter/doc/freediameter
    .conf.sample /usr/local/etc/freeDiameter/freeDiameter.conf
2 root@diamserver:~# mkdir -p /usr/local/etc/freeDiameter/extensions/samples/
3 root@diamserver:~# cp -r /install_files/freediameter/freeDiameter/doc/. /usr/
    local/etc/freeDiameter/extensions/samples/
```

Výpis 27: Přesunutí vzorových konfiguračních souborů do pracovního adresáře

7.4 Definice a zprovoznění účastníků

Pro základní definici účastníků, tedy Diameter serveru a Diameter klienta, je potřebné konfigurovat pouze několik síťových parametrů a cesty k vytvořeným certifikátům. Definice účastníků se provede úpravou hlavního konfiguračního souboru *freeDiameter.conf* dle výpisu 28 a nyní konkrétně z pohledu virtuálního stroje *diamserver*.

```
1 TLS_Cred = "/usr/local/etc/freeDiameter/diametercert/cert.pem", "/usr/local/etc
    /freeDiameter/diametercert/privkey.pem";
2 TLS_CA = "/usr/local/etc/freeDiameter/diametercert/cert.pem";
3 TLS_DH_File = "/usr/local/etc/freeDiameter/diametercert/dh.pem";
4
5 Identity = "diamserver.vsb.cz";
6 Realm = "vsb.cz";
7
8 ConnectPeer = "diamclient.vsb.cz" { No_TLS; };
```

Výpis 28: Základní definice účastníků v konfiguračním souboru *freeDiameter.conf*

Ve výpisu 28 můžete nejprve vidět přiřazení cesty k privátnímu certifikátu, certifikační autoritě a souboru, který definuje výměnu klíčů. Identita daného lokálního účastníka musí odpovídat běžnému jménu privátního certifikátu a hodnotě FQDN, kterou má přiřazenou aktuálně přihlášený hostitel virtuálního stroje. Pro kontrolu správně definovaného názvu hostitele a FQDN

můžete použít informace zahrnuté v kapitole 5.3. Využitím parametru *ConnectPeer* lze definovat různý počet účastníků či agentů, s kterými je virtuální stroj v komunikaci. Virtuální stroj *diamclient*, respektive Diameter klient, obsahuje totožnou konfiguraci, která se pouze odlišuje zrcadlením hodnoty parametru *Identity* a *ConnectPeer*.

Po provedení základní konfigurace této kapitoly je možné spustit program freeDiameter v ladicím režimu na obou virtuálních strojích *diamserver* a *diamclient* užitím příkazu z výpisu 29. Daný výpis dále obsahuje textový výstup programu freeDiameter obsahující úspěšné navázání komunikace s účastníkem *diamclient.vsb.cz*, který je umístěn na virtuálním stroji *diamclient*. Před navázáním komunikace proběhla výměna Diameter zpráv *Capabilities-Exchange-Request* a *Capabilities-Exchange-Answer*, které značí vyjednávání schopností mezi účastníky. Podpora pro vyjednání schopností byla popsána v kapitole 4.9.4 a je vyhrazena pouze protokolu Diameter. Komunikace mezi účastníky byla zřízena bez zabezpečení TLS na základě definice parametru *No_TLS* mezi účastníky a využitím již sebou podepsaných certifikátů.

```
1  root@diamserver:~# freeDiameterd -dd
2
3  10:43:39 NOTI  Local server address(es): 192.168.1.115{---L-}
4  10:43:39 NOTI  freeDiameterd daemon initialized.
5  10:43:39  DBG  diamclient.vsb.cz: Connecting...
6  10:43:39  DBG  'STATE_CLOSED' -> 'STATE_WAITCNXACK' 'diamclient.vsb.cz'
7  10:43:39  DBG  Connecting to SCTP 192.168.1.116(0):3868...}
8  10:43:39  DBG  SENT to 'diamclient.vsb.cz': 'Capabilities-Exchange-Request'
9  10:43:39  DBG  'STATE_WAITCNXACK' -> 'STATE_WAITCEA' 'diamclient.vsb.cz'
10 10:43:39 NOTI  Connected to 'diamclient.vsb.cz' (SCTP,soc#21), remote
       capabilities:
11 10:43:39 NOTI      Capabilities-Exchange-Answer(257)[----]
12 10:43:39  DBG  diamclient.vsb.cz: Connection established, {----} SCTP,#
       21->192.168.1.116(3868)
13 10:43:39 NOTI  No TLS protection negotiated with peer 'diamclient.vsb.cz'.
14 10:43:39 NOTI  'STATE_WAITCEA' -> 'STATE_OPEN' 'diamclient.vsb.cz'
```

Výpis 29: Spuštění programu freeDiameter v ladícím režimu s textovým výstupem

7.5 Zásuvné moduly programu freeDiameter

Program freeDiameter obsahuje rozsáhlou řadu zásuvných modulů, mezi které patří i různé aplikace protokolu Diameter, jejichž koncept byl objasněn v kapitole 4.1. Zásuvné moduly či aplikace složí pro rozšíření základní specifikace protokolu Diameter například pro implementaci síťového směrování, řízení účastníků, slovníkových překladačů, ladících nástrojů a mnoha dalších služeb.

Zásuvný modul je sdílený objekt s typickým názvem přípony *.fdx* a výchozím úložištěm v adresáři */usr/local/lib/freeDiameter/*. V kapitole 7.1 byla zvolena instalace programu freeDiameter se všemi dostupnými zásuvnými moduly a obsah výchozího adresáře zásuvných modulů je tedy úplný.

Jednotlivé zásuvné moduly rozdělit dle využití na základě jejich prefixů následovně:

- *dict_*: Označuje zásuvné moduly k definici obsahu určitých slovníků.
- *dbg_*: Moduly s tímto prefixem slouží pouze pro výpis dodatečných informační o chodu programu freeDiameter.
- *acl_*: Tyto zásuvné moduly řídí síťový přístup jednotlivých účastníků.
- *rt_*: Pro služby směrování je využito těchto zásuvných modulů.
- *app_*: Jedná se o aplikace protokolu Diameter, které se typicky zabývají zaznamenáním a zpracováním specifických zpráv.
- *test_*: Posledním typem jsou zásuvné moduly využitelné pro ověření funkčnosti nasazené implementace.

K části zásuvných modulů je přiřazen i konfigurační soubor, který je v určitých případech nutné upravit pro zajištění správné funkčnosti. Konfigurační soubory jednotlivých zásuvných modulů jsou umístěny v adresáři */extensions/*. Zajištění spuštění několika zásuvných modulů při inicializaci programu freeDiameter je znázorněno ve výpisu 30 způsobem úpravy konfiguračního souboru *freeDiameter.conf*.

```
1 LoadExtension = "dict_eap.fdx";
2 LoadExtension = "test_app.fdx" : "/usr/local/etc/freeDiameter/extensions/
    test_app.conf";
3 LoadExtension = "app_diameap.fdx" : "/usr/local/etc/freeDiameter/extensions/
    app_diameap.conf";
```

Výpis 30: Vzorová definice zásuvného modulu v konfiguračním souboru *freeDiameter.conf*

7.6 Ověření dostupnosti mezi účastníky

Před rozšiřováním implementace protokolu Diameter je nejprve vhodné ověřit možnost komunikace mezi jednotlivými Diameter účastníky. Pro tento účel existuje vymezený zásuvný modul *test_app.fdx*, jehož konfiguračním souborem je */extensions/test_app.conf*. Funkce zásuvného modulu *test_app.fdx* je velmi podobná unixovému příkazu *ping*. Konfigurace inicializace zásuvného modulu se provede úpravou konfiguračního souboru *freeDiameter.conf* dle výpisu

30 z předchozí kapitoly. Rozdíl v obsahu přiřazeného konfiguračního souboru */extensions/-test_app.conf* mezi virtuálním strojem *diamserver* a *diamclient* spočívá ve změně tvaru parametru na *mode=server* nebo *mode=client*.

Součástí instalace programu freeDiameter bylo i vytvoření adresáře */scripts/*, který obsahuje několik skriptovacích souborů stažených ze stránek vývojáře programu freeDiameter [25]. Většina těchto skriptovacích souborů slouží pro instalaci programu freeDiameter na jiné operační systémy či instalaci některých síťových služeb. Do adresáře jsem dodatečně přidal skriptovací soubor *ping_app.sh*, jehož úplný obsah můžete vidět ve výpisu 31. Tento skriptovací soubor umožňuje zaslat opakovaný dotaz na všechny účastníky v doméně lokálního účastníka (*vsb.cz*), na který všichni dostupní účastníci odpoví. Tímto způsobem lze ověřit funkčnost i složitějších síťových návrhů se směrovacími službami při úpravně konfiguračního souboru */extensions/test_app.conf*.

```
1 #!/bin/bash -x
2
3 PID=‘ps --no-heading -C freeDiameterd -o %p‘;
4 if [ "$1" = "loop" ];
5 then
6   while (true); do sleep 10; kill -USR1 $PID; done;
7 else
8   kill -USR1 $PID;
9 fi
```

Výpis 31: Obsah skriptovacího souboru *ping_app.sh*

Příkaz pro spuštění skriptovacího souboru společně s textovým výstupem programu freeDiameter v ladícím režimu můžete vidět ve výpisu 32. Součástí textového výstupu je FQDN účastníka, který úspěšně přijal komunikační dotaz a odeslal odpověď. Na závěr je vypsána statistika odeslaných, chybných a přijatých zpráv.

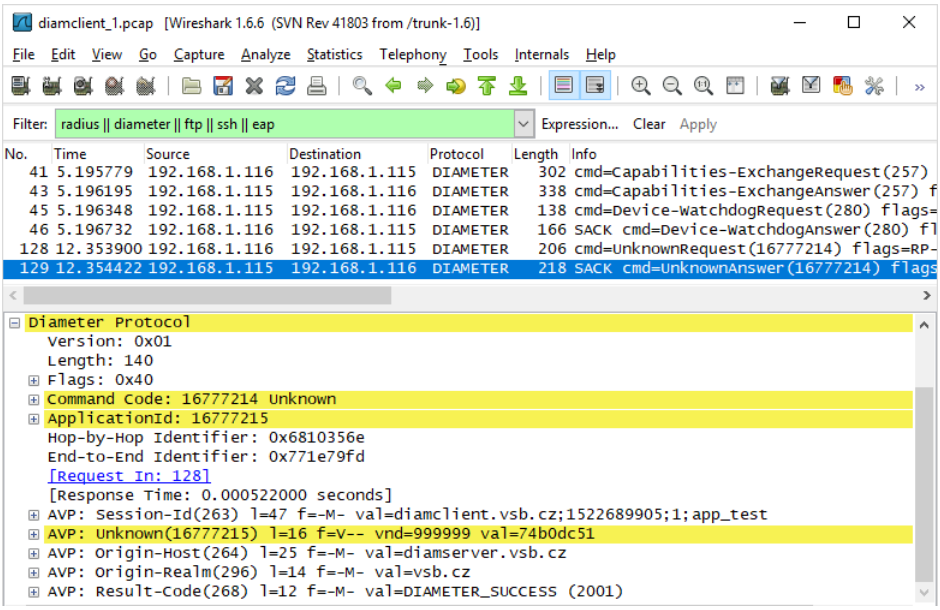
Analýza zachycené komunikace souboru *diamclient_1.pcap* v prostředí Wireshark je vyobrazena na obrázku 7.6.1. V obrázku můžete vidět výměnu Diameter zpráv mezi IP adresami 192.168.1.115 (*diamserver*) a 192.168.1.116 (*diamclient*). V kapitole 3.2.1 již bylo řečeno, že typ zprávy protokolu RADIUS či Diameter je definovaný hodnotou pole kódu (angl. command code). V zachycené komunikaci můžete vidět, že došlo k přenosu celkem tří typů Diameter zpráv s odpovídajícím kódem. Diameter zpráva *Capabilities-Exchange* obsahuje vyjednání schopností mezi účastníky. Navazující zpráva *Device-Watchdog* je speciální hlídací mechanismus vymezený protokolu Diameter a popsaný v kapitole 4.9.1. Poslední typ zprávy byl programem Wireshark označený za *UnknownRequest*, jelikož kód Diameter zprávy není obsažený ve slovníku programu Wireshark. Program freeDiameter automaticky rozpozná daný kód a Diameter zprávu označí za *Test-Request* a to nativně bez nutnosti inicializace dodatečného zásuvného modulu slovníku. Nasazením specifických Diameter aplikací je možné definovat nové kódy a atributy, které jsou

přenášeny Diameter zprávami. Tento samotný fakt přináší výrazně vyšší rozšiřitelnost funkčnosti oproti protokolu RADIUS, jehož maximální počet možných atributů je 255.

```
root@diamclient:# sh /usr/local/etc/freeDiameter/scripts/ping_app.sh

SEND 44344c22 to 'vsb.cz' (-)
13:24:19  DBG   SENT to 'diamserver.vsb.cz': 'Test-Request'16777215/16777214
13:24:19  DBG   RCV from 'diamserver.vsb.cz': (no model)16777215/16777214 f:-P
    -- src:'diamserver.vsb.cz'
13:24:19  DBG   DISPATCHING: (no model)16777215/16777214
RECV 44344c22 (Ok) Status: 2001 From 'diamserver.vsb.cz' ('vsb.cz') in 0.028732
     sec
13:24:28  DBG   ------- app_test statistics ---------
13:24:28  DBG    Executing for: 6526.257255 sec
13:24:28  DBG    Client:
13:24:28  DBG      1 message(s) sent
13:24:28  DBG      0 error(s) received
13:24:28  DBG      1 answer(s) received
13:24:28  DBG        fastest: 0.028732 sec.
13:24:28  DBG        slowest: 0.028732 sec.
13:24:28  DBG        Average: 0.028732 sec.
13:24:28  DBG   ------------------------------------
```

Výpis 32: Ověření dostupnosti mezi účastníky využitím zásuvného modulu *test_app.fdx*



Obrázek 7.6.1: Analýza záznamu komunikace *diamclient_1.pcap* v prostředí Wireshark

7.7 Konfigurace autentizace užitím metod EAP

Samotný program freeDiameter zajišťuje pouze podporu základní specifikace protokolu Diameter, zatímco funkčnost je primárně rozšiřitelná instalací a konfigurací zásuvných modulů a aplikací. Pro podporu autentizace uživatelů užitím různých metod EAP je možné využít projektu DiamEAP, jehož popis lze nalézt na internetových stránkách vývojáře [27].

Jelikož rozhodování o autentizaci uživatelů vždy vyhodnocuje AAA server, bude navazující konfigurace nasazena na virtuální stroj *diamserver*. K autentizaci uživatele určitým typem metody EAP je nutné konfigurovat aplikaci *app_diameap.fdx*, jenž již byla nainstalována v kapitole 7.1. Před samotnou konfigurací dané aplikace je nejprve nutné nastavit MYSQL databázi a naplnit ji uživatelskými údaji dle výpisu 33. Ve výpisu se nejprve vytvořila databáze s názvem *diameap_ui* s datovou strukturou dle schématu ze souboru */extensions/app_diameap/diameap.sql*. Databáze obsahuje celkem tři uživatele s preferovaným typem metody EAP-MD5 (4) a heslo pro přístup do databáze je *diameap_heslo*.

```
1 root@diamserver:# mysqladmin -u root -p create diameap_ui
2 root@diamserver:# mysql -u root -p diameap_ui < /usr/local/etc/freeDiameter/
    extensions/app_diameap/diameap.sql
3 root@diamserver:# echo "INSERT INTO users (username, password, eapmethod)
    values ( 'petr_diameap', 'petr_diameap_heslo', 4 );" | mysql -u root -p
    diameap_ui
4 root@diamserver:# echo "INSERT INTO users (username, password, eapmethod)
    values ( 'marie_diameap', 'marie_diameap_heslo', 4 );" | mysql -u root -p
    diameap_ui
5 root@diamserver:# echo "INSERT INTO users (username, password, eapmethod)
    values ( 'martina_diameap', 'martina_diameap_heslo', 4 );" | mysql -u root
    -p diameap_ui
6 root@diamserver:# mysqladmin -u root password diameap_heslo
```

Výpis 33: Nastavení a naplnění MYSQL databáze k využití aplikace *app_diameap.fdx*

Konfigurační soubor */extensions/app_diameap.conf* slouží k nastavení Diameter aplikace *app_diameap.fdx* a musí obsahovat údaje dle výpisu 34.

```
1 DiamEAP_MySQL = "root" , "diameap_heslo" , "localhost" , "diameap_ui";
2 Load_plugin = "EAP Identity":1:0:"eap_identity.emp":"";
3 Load_plugin = "EAP MD5":4:0:"eap_md5.emp":"";
```

Výpis 34: Konfigurace aplikace *app_diameap.fdx* v souboru */extensions/app_diameap.conf*

7.8 Překladatelský agent RADIUS a Diameter zpráv

V kapitole 4.10 byl popsán koncept překladatelského agenta, jehož účelem je zajistit interoperabilitu mezi RADIUS a Diameter systémy způsobem překladu zpráv. Pro konfiguraci překladatelského agenta lze využít aplikace *app_radgw.fdx*, jenž je umístěna na virtuálním stroji *diamserver*.

Představou je, aby Diameter klient překládal přijaté RADIUS zprávy a komunikoval s Diameter serverem pouze užitím zpráv protokolu Diameter. Tato představa nebyla v mém řešení realizovatelná z důvodu nežádoucí interakce mezi aplikací *app_radgw.fdx* a jinými službami. Z tohoto důvodu je překladatelský agent umístěný na straně Diameter serveru. Pro konfiguraci aplikace je nutné upravit konfigurační soubor */extensions/app_radgw.conf* dle výpisu 35. Konfigurace zahrnuje definici dodatečných zásuvných modulů a určení IP adresy s portem autentizačního i účtovacího serveru (lokální hostitel). Dále konfigurace obsahuje definici IP adresy zařízení NAS společně se sdíleným heslem, jenž zasílá RADIUS zprávy k překladu.

```
1  RGWX = "echodrop.rgwx" : "/usr/local/etc/freeDiameter/extensions/echodrop.rgwx.
       conf";
2  RGWX = "auth.rgwx" : auth;
3  RGWX = "acct.rgwx" : acct;
4  RGWX = "debug.rgwx";
5
6  auth_server_enable = 1;
7  auth_server_port = 1812;
8  auth_server_ip4 = 0.0.0.0;
9  acct_server_enable = 1;
10 acct_server_port = 1813;
11 acct_server_ip4 = 0.0.0.0;
12
13 nas = 192.168.1.116 / "sharedpass123";
```

Výpis 35: Konfigurace překladatelského agenta v souboru */extensions/app_radgw.conf*

7.9 Konfigurace Diameter klienta

V kapitole 7.4 byl na virtuální stroj *diamclient* nainstalovaný pouze program freeDiameter, jenž zajišťuje podporu základní specifikace protokolu Diameter mezi všemi komunikujícími účastníky. Na tento virtuální stroj byly dále konfigurovány některé slovníkové a informační zásuvné moduly programu freeDimeter, které výrazně nerozšiřují samotnou funkčnost.

Konfigurace Diameter klienta dále obsahuje instalaci simulačního nástroje *freeradius-utils* a konfiguraci programu Hostapd, kterým byl zřízen přístupový bod při návrhu implementace protokolu RADIUS v kapitole 6.10. Konfigurace programu Hostapd na virtuálním serveru *diamc-*

lient je obecně shodná s konfigurací dle výpisu 19. V implementaci protokolu Diameter nebyly instalovány služby SSH a FTP, jelikož v době psaní práce neexistovaly zásuvné autentizační moduly (PAM), které umožňovaly generovat Diameter zprávy.

7.10 Žádost o přístup od AAA uživatele

Postup zprovoznění AAA uživatele byl již popsán v kapitole 6.11. Pro implementaci Diameter protokolu byl vytvořený nový konfigurační soubor programu wpa_supplicant dle výpisu 36. Definované uživatelské údaje a typ metody EAP-MD5 odpovídají obsahu MYSQL databáze, která byla naplněna v kapitole 7.7.

```
network={
   ssid="diameter.ap.vsb.cz"
   key_mgmt=WPA-EAP
   pairwise=TKIP
   group=TKIP
   eap=MD5
   identity="petr_diameap"
   password="petr_diameap_heslo"
   phase1="peapver=0"
   phase2="MSCHAPV2"
}
```

Výpis 36: Konfigurace programu wpa_supplicant pro implementaci protokolu Diameter

Příkaz pro spuštění programu wpa_supplicant z virtuálního stroje *aaauser* obsahuje výpis 37. Z výpisu je zřejmé, že proces autentizace užitím metody EAP započal, ale uživatel nebyl úspěšně autentizován vůči Diameter serveru.

```
root@aaauser:~# wpa_supplicant -i eth0 -c /etc/wpa_supplicant/diameter_vsb_wpa.
    conf -D wired

Successfully initialized wpa_supplicant
eth0: Associated with 01:80:c2:00:00:03
eth0: CTRL-EVENT-EAP-STARTED EAP authentication started
eth0: CTRL-EVENT-EAP-FAILURE EAP authentication failed
```

Výpis 37: Autentizace vůči Diameter serveru užitím programu wpa_supplicant

Textový výstup programu Hostapd, který je umístěný na virtuálním stroji *diamclient*, je obsažený ve výpisu 38. Z výpisu lze zjistit, že program Hostapd inicializoval přístupový bod, vyjednal komunikaci s programem wpa_supplicant užitím typu metody EAP-MD5 (zde značeno číslem 1) a vyhodnocení autentizace společně s účtováním předal vůči virtuálnímu stroji *diamserver* s IP adresou 192.168.1.115.

```
root@diamclient:~# hostapd /etc/hostapd/hostapd.conf

Configuration file: /etc/hostapd/hostapd.conf
Using interface eth0 with hwaddr 00:0c:29:ad:fe:d1 and ssid ""
eth0: RADIUS Authentication server 192.168.1.115:1812
eth0: RADIUS Accounting server 192.168.1.115:1813
eth0: interface state UNINITIALIZED->ENABLED
eth0: AP-ENABLED
eth0: CTRL-EVENT-EAP-STARTED 00:0c:29:37:b5:15
eth0: CTRL-EVENT-EAP-PROPOSED-METHOD vendor=0 method=1
```

Výpis 38: Textový výstup programu Hostapd při autentizaci uživatele vůči Diameter serveru

Částečný textový výstup programu freeDiameter v ladícím režimu obsahuje výpis 39. Z textového výstupu je možné identifikovat problém neúspěšné autentizace uživatele *petr_diameap* užitím programu *wpa_supplicant*. V kapitole 7.8 byla již provedena konfigurace Diameter aplikace *app_radgw.fdx*, jenž funguje jako překladatelský agent mezi RADIUS a Diameter zprávami. Textový výpis zahrnuje přijetí RADIUS zprávy Žádost-Přístupu (angl. *Access-Request*) a nedokončené vytvoření nové Diameter zprávy odpovídajících AVP. Aplikace překladatelského agenta nedokázala zpracovat AVP typu *Acct-Session-Id* obsažený v původní RADIUS zprávě z důvodu chybějící podpory a překlad mezi protokoly se přerušil. Ve výpisu 35 byl taktéž definovaný zásuvný modul pro překlad účtovacích zpráv, ale výsledek překladu byl stále neúspěšný.

```
13:12:30  DBG   RADIUS: RCV 196B from 192.168.1.116(55048)
13:12:30  DBG   ------------- RADIUS/Diameter Request Debug -------------
13:12:30  DBG    RADIUS request (0xb4100c20) DUMP:
13:12:30  DBG    id  : 0x03, code: 1 (Access-Request [RFC2865])
13:21:07  DBG    Diameter message (0xb21014b8) DUMP:
   AVP: 'User-Name'(1) l=8 f=-M val="petr_diameap"
   AVP: 'Origin-Host'(264) l=8 f=-M val="diamclient.vsb.cz"
   AVP: 'NAS-Identifier'(32) l=8 f=-M val="diameter.ap.vsb.cz"
   ...
13:12:30  DBG   ============ Debug complete =============
13:12:30  DBG   [radgw] No plugin available to handle attribute 44 (Acct-
    Session-Id [RFC2866]) in command 1 (Access-Request [RFC2865])! Translation
    aborted.
13:12:30  NOTI   1 problem(s) occurred while translating a RADIUS message, data
     discarded.
```

Výpis 39: Část textového výstupu programu freeDiameter při neúspěšné autentizaci uživatele

Dále jsem ověřoval funkčnost autentizace pomocí simulačního nástroje *radtest*, který byl již využitý v návrhu implementace protokolu RADIUS v kapitole 6.6. Ze strany Diameter klienta *diamclient* byl zadán příkaz dle výpisu 40, který dále obsahuje textový výstup nástroje *radtest*. Jelikož nástroj *radtest* slouží pro interní ověření funkčnosti s programem freeRADIUS, jedná program freeDiameter se zprávou jako zástupce. Žádost o přístup je zamítnuta, jelikož program freeDiameter nedokáže zprávu sám vyhodnotit a snaží se ji předat bez dostupné podpory.

```
root@diamclient:~# radtest petr_diameap petr_diameap_heslo diamserver 10
    sharedpass123

Sending Access-Request of id 113 to 192.168.1.115 port 1812
User-Name = "petr_diameap"
User-Password = "petr_diameap_heslo"
NAS-IP-Address = 192.168.1.116
NAS-Port = 10
Message-Authenticator = 0x00000000000000000000000000000000
rad_recv: Access-Reject packet from host 192.168.1.115 port 1812, id=113,
    length=91
Reply-Message = "No suitable candidate to route the message to"
Error-Cause = Proxy-Request-Not-Routable
Message-Authenticator = 0xf6f4d31f2417c222566b00f12e4a8895
```

Výpis 40: Ověření funkčnosti autentizace uživatele nástrojem *radtest*

Důvodem neúspěšné autentizace uživatelů je, že neexistují licenčně otevřené programy k zprovoznění klientské části Diameter protokolu. Přístupový bod programu Hostapd zajišťuje podporu pouze autentizačního a účtovacího serveru typu RADIUS. Dále neexistuje řešení zásuvných autentizačních modulů (PAM), jenž jsou schopné generovat zprávy protokolu Diameter. V případě dostupné podpory protokolu Diameter by nebylo nutné vyžívat nespolehlivých překladatelských agentů pro převod zpráv mezi protokoly.

7.11 Porovnání mezi implementací protokolů RADIUS a Diameter

Pro návrh serverové části protokolu RADIUS byl využitý program FreeRADIUS, jehož domovské stránky jsou zde [22]. FreeRADIUS je licenčně otevřený, velmi rozšířený a aktivně se rozvíjející projekt. Na projektu FreeRADIUS stále pracuje rozsáhlá komunita programátorů a síťových návrhářů, jak je možné vidět z velkého množství publikované literatury a internetových fór. Pro návrh klientské části protokolu RADIUS existuje rozsáhlá řada licenčně otevřených implementací, mezi které například patří využití PAM pro zprostředkování přístupu k určité službě (SSH, FTP a další) nebo instalace přístupového bodu programem Hostapd. Protokol RADIUS obsahuje řadu nedostatků z oblasti spolehlivosti a zabezpečení, které již byly zmíněny v kapitolách

3.4 a 3.5.1. I přes tyto existující nedostatky je program FreeRADIUS stále populárně využíván jako robustní a modulárně rozšiřitelná implementace serverové části protokolu RADIUS.

Pro návrh protokolu Diameter jsem použil program freeDiameter, o kterém se můžete informovat z domovských stránek daného projektu [25]. Program freeDiameter je taktéž licenčně otevřený a funkčně rozšiřitelný způsobem instalace specifických zásuvných modulů a Diameter aplikací. Při návrhu serverové části lze využít podpory uživatelské autentizace několika typů metod EAP a dále podpory překladatelského agenta zpráv mezi protokoly RADIUS a Diameter. Program freeDiameter nezajišťuje podporu pro starší protokoly autentizace jako PAP a CHAP, kterými disponuje program FreeRADIUS. V kapitole 7.10 bylo již řečeno, že neexistují licenčně otevřené implementace klientské části protokolu Diameter. Z tohoto důvodu nebylo proveditelné zajistit úspěšnou autentizaci uživatelů, kteří žádali o přístup.

V případě budoucího navázání na téma problematiky implementace protokolů RADIUS a Diameter bych čtenářovi doporučil, aby přednostně věnoval svou pozornost vývoji licenčně otevřených implementací klientské části protokolu Diameter. Do klientské části spadají PAM schopné generovat Diameter zprávy a program Hostapd. V různých internetových diskuzích již nyní cirkulují informace o vývoji programu Hostapd s podporou protokolu Diameter. Dále bude v polovině letošního roku 2018 publikována literatura [28], která se bude zabývat návrhem a praktickým nasazením protokolu Diameter s využitím projektu freeDiameter. Při rostoucím počtu nových bezdrátových a mobilních technologií je rozumné očekávat, že vývoj implementací protokolů RADIUS a Diameter bude dále pokračovat.

8 Závěr

Cílem diplomové práce bylo provést návrh autentizace a autorizace uživatelů s využitím protokolů RADIUS a Diameter v prostředí virtuálních serverů. Funkčnost implementace byla ověřena pod operačním systémem Ubuntu 14.04.5 LTS na fyzických i virtuálních strojích a je plně funkční pro řešení s IPv4 a IPv6.

K návrhu implementace protokolu RADIUS byl využitý program FreeRADIUS 2.2.10, jehož síťové schéma obsahovalo dva RADIUS servery různých domén, dva RADIUS klienty a jednoho uživatele. Řešení zahrnovalo konfiguraci databází MYSQL a LDAP pro uložení uživatelských údajů a účtovacích dat. Implementace obsahovala funkční řešení pomocí protokolů autentizace PAP, CHAP a různých typů metod EAP (EAP-MD5, EAP-TTLS, EAP-PEAP a dalších). Pro autentizaci a autorizaci uživatelů bylo dále využito zásuvných autentizačních modulů (PAM) pro udělení přístupu k instalovaným službám SSH a FTP. Uživatelé byli taktéž autentizování při připojením vůči přístupovému bodu programu Hostapd, jenž umožňoval zprostředkovat zabezpečenou komunikaci se suplikantem wpa_supplicant.

Pro návrh implementace protokolu Diameter bylo využito programu freeDiameter 1.2.1 a síťové schéma se skládalo z virtuálního stroje Diameter serveru, Diameter klienta a jednoho uživatele. Základní specifikace protokolu Diameter byla v řešení rozšířena o podporu autentizace uživatelů užitím různých typů metod EAP a možnosti překladu zpráv mezi protokoly RADIUS a Diameter. Samotná funkční autentizace uživatelů nebyla proveditelná z důvodu, že neexistuje licenčně otevřené řešení klientské části protokolu Diameter a to například v podobě PAM či programu Hostapd.

Porovnání a analýza protokolů RADIUS a Diameter byla provedena v teoretické i praktické části. Protokoly byly porovnány z hlediska struktury zpráv, úrovně zabezpečení, spolehlivosti, rozšiřitelnosti a vzájemné interakce. Analýza praktického implementace protokolů RADIUS a Diameter byla následně provedena rozborem síťové komunikace v prostředí Wireshark a využitím několika simulačních nástrojů.

Na závěr je výstup diplomové práce umístěný na přiloženém CD a obsahuje řadu skriptovacích souborů a záznamů zachycené síťové komunikace. Skriptovací soubory lze využít pro sestavení úplné implementace protokolů RADIUS a Diameter, která byla provedena v praktické části práce. Skriptovací soubory dále obsahují textové komentáře rozvíjející řešenou problematiku a sadu číslovaných simulačních příkazů pro ověření funkčnosti řešení. Přiložené záznamy síťové komunikace je možné spustit a dále analyzovat v prostředí Wireshark.

Literatura

[1] CARROLL, Brandon. *Cisco access control security: AAA administrative services.* Indianapolis, Ind.: Cisco Press, 2004. ISBN 1-58705-124-9.

[2] NAKHJIRI, Madjid. a Mahsa. NAKHJIRI. *AAA and network security for mobile access: radius, diameter, EAP, PKI and IP mobility.* Hoboken, NJ: John Wiley, 2005. ISBN 978-0-470-01194-2.

[3] *Generic AAA Architecture.* [online]. IETF, RFC 2903 [cit. 2017-12-26]. Dostupný z WWW: <https://tools.ietf.org/html/rfc2903>.

[4] *EAP-TLS Autentication Protocol.* [online]. IETF, RFC 5216 [cit. 2017-12-28]. Dostupný z WWW: <https://tools.ietf.org/html/rfc5216>.

[5] HASSELL, Jonathan. *RADIUS.* Sebastopol: O'Reilly, 2003. ISBN 0-596-00322-6.

[6] *Remote Authentication Dial In User Service (RADIUS).* [online]. IETF, RFC 2865 [cit. 2017-12-29]. Dostupný z WWW: <https://tools.ietf.org/html/rfc2865>.

[7] *RADIUS (Remote Authentication Dial In User Service) Support For Extensible Authentication Protocol (EAP).* [online]. IETF, RFC 3579 [cit. 2018-01-16]. Dostupný z WWW: <https://www.ietf.org/rfc/rfc3579>.

[8] *RADIUS Accounting.* [online]. IETF, RFC 2866 [cit. 2018-01-16]. Dostupný z WWW: <https://tools.ietf.org/html/rfc2866>.

[9] *Roaming Operations.* [online]. IETF, roamops working group [cit. 2018-01-16]. Dostupný z WWW: <https://datatracker.ietf.org/wg/roamops/documents/>.

[10] *Proxy Chaining and Policy Implementation in Roaming.* [online]. IETF, RFC 2607 [cit. 2018-01-17]. Dostupný z WWW: <https://tools.ietf.org/html/rfc2607>.

[11] *RADIUS EXTensions.* [online]. IETF, radext working group [cit. 2018-01-17]. Dostupný z WWW: <https://tools.ietf.org/wg/radext/>.

[12] *Criteria for Evaluating AAA Protocols for Network Access.* [online]. IETF, RFC 2989 [cit. 2018-01-17]. Dostupný z WWW: <https://tools.ietf.org/search/rfc2989>.

[13] *Authentication, Authorization, and Accounting: Protocol Evaluation.* [online]. IETF, RFC 3127 [cit. 2018-01-17]. Dostupný z WWW: <https://tools.ietf.org/html/rfc3127>.

[14] CHEN, Jyh-Cheng a Tao ZHANG. *IP-based next-generation wireless networks: systems, architectures, and protocols.* Hoboken, N.J.: Wiley-Interscience, 2004. ISBN 0-471-23526-1.

[15] *Diameter Base Protocol (September 2003).* [online]. IETF, RFC 3588 [cit. 2018-02-17]. Dostupný z WWW: <https://tools.ietf.org/html/rfc3588>.

[16] *Diameter Base Protocol (October 2012).* [online]. IETF, RFC 6733 [cit. 2018-02-17]. Dostupný z WWW: <https://tools.ietf.org/html/rfc6733>.

[17] *VMware Workstation Pro 14 ke stažení.* [online]. WMware, Inc [cit. 2018-03-03]. Dostupný z WWW: <https://www.vmware.com/products/workstation-pro/workstation-pro-evaluation.html>.

[18] *Ubuntu 14.04.5 LTS (Trusty Tahr) ke stažení.* [online]. Ubuntu Project [cit. 2018-03-03]. Dostupný z WWW: <http://releases.ubuntu.com/14.04/>.

[19] *Skritovací soubory ke stažení.* [online]. [cit. 2018-03-08]. Dostupný z WWW: <https://pastebin.com/8dtR8AP1>.

[20] *JRadius Simulator nástroj ke stažení.* [online]. Coova [cit. 2018-03-09]. Dostupný z WWW: <https://github.com/coova/jradius/releases>.

[21] *Notepad++ program ke stažení.* [online]. [cit. 2018-03-09]. Dostupný z WWW: <https://notepad-plus-plus.org/download/v7.5.6.html>.

[22] *Internetové stránky projektu FreeRADIUS.* [online]. The FreeRADIUS Server Project [cit. 2018-03-11]. Dostupný z WWW: <https://freeradius.org/releases/>.

[23] Walt, Dirk. *FreeRADIUS beginner's guide : manage your network resources with FreeRADIUS.* Birmingham U.K: Packt Pub, 2011. ISBN 978-1849514088.

[24] *Wireshark prostředí ke stažení.* [online]. [cit. 2018-03-22]. Dostupný z WWW: <https://www.wireshark.org/download.html>.

[25] *Internetové stránky projektu freeDiameter.* [online]. [cit. 2018-03-23]. Dostupný z WWW: <http://www.freediameter.net/trac/blog/about>.

[26] *Zdrojové soubory programu freeDiameter.* [online]. [cit. 2018-03-23]. Dostupný z WWW: <https://github.com/Metaswitch/freeDiameter>.

[27] *Internetové stránky projektu DiamEAP.* [online]. The DiamEAP Project team [cit. 2018-03-26]. Dostupný z WWW: <http://diameap.yagami.freediameter.net/>.

[28] Hannes Tschofenig and Sebastien Decugis and Jean Mahoney and Jouni Korhonen. *Diameter: New Generation AAA Protocol - Design, Practice and Applications.* Wiley, 2018. ISBN 1118875907.

Seznam příloh

Součástí diplomové práce je pouze přiložené CD, jehož adresářová struktura je:

- Kořenový adresář
  - JANCA_Lukas_DP_2018.pdf
  - /skriptovaci_soubory/
    - hostname_setup.sh
    - freeradius_install_server.sh
    - freeradius_install_client.sh
    - freeradius_install_proxy.sh
    - freeradius_install_access_point.sh
    - freediameter_install_server.sh
    - freediameter_install_client.sh
    - aaa_install_wpa_supplicant.sh
  - /wireshark_zaznamy/
    - radiusclient_1.pcap
    - radiusclient_....pcap
    - radiusclient_19.pcap
    - radiusaccesspoint_1.pcap
    - radiusaccesspoint_2.pcap
    - diamclient_1.pcap
    - diamclient_2.pcap
    - aaauser_1.pcap
    - aaauser_....pcap
    - aaauser_5.pcap